Title

An Information-Theoretic Exploration of Generalization and Data Value

ISBN

9798288858505

Author

Rammal, Mohamad Rida

Publication Date

2025-07-14

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

An Information-Theoretic Exploration of Generalization and Data Value

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical and Computer Engineering

by

Mohamad Rida Rammal

2025

ABSTRACT OF THE DISSERTATION

An Information-Theoretic Exploration of Generalization and Data Value

by

Mohamad Rida Rammal
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2025
Professor Suhas N. Diggavi, Chair

In recent years, there has been a strong push towards developing increasingly large neural networks, often with billions of parameters and trained on massive datasets, due to their impressive performance in various machine learning tasks. Despite their widespread success, our understanding of why these models perform so well, and which data contribute the most to that performance, remains limited. Information theory offers foundational tools for quantifying the information content in data, framing learning as the process of extracting that information. Given this close relationship, we investigate two key problems from an information-theoretic perspective: (1) the generalization of machine learning algorithms and (2) data valuation for generative models. Specifically, we introduce a new class of generalization bounds based on the leave-one-out conditional mutual information, study its properties, and evaluate it in practical scenarios. Moreover, we propose a novel data valuation function for language models, based on intuitive ideas of what makes data valuable and inspired by distribution testing. We demonstrate that our proposed function is computationally efficient, possesses several desirable theoretical properties, and evaluate it on real-world datasets.

The dissertation of Mohamad Rida Rammal is approved.

Lin Yang

Jonathan C. Kao

Stefano Soatto

Suhas N. Diggavi, Committee Chair

University of California, Los Angeles

2025

To my parents and sister, whose love and support know no bounds, and to my brother Hadi, who has been a constant pillar of strength throughout.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my advisor Professor Suhas Diggavi. I truly believe I wouldn't have reached this far without his immense guidance, support, and patience.

I would like to thank my collaborators: Ruida Zhou, Stefano Soatto, Alessandro Achille, Ashutosh Sabharwal, and Aditya Golatkar. My experiences working with them have made me a more competent researcher.

I would also like to extend my gratitude to Professors Louay Bazzi, Ibrahim Abou-Faycal, and Kamal Khuri Makdisi from AUB. They instilled in me a love for mathematics, and it was Professor Bazzi who introduced me to my future PhD advisor.

Vita

2019	B.S. Computer and Communication Engineering, American University of Beirut, Lebanon.
2019	B.S. Mathematics, American University of Beirut, Lebanon.
2021	M.S. Electrical and Computer Engineering, UCLA, Los Angeles, California.
2022	Applied Scientist Intern, Amazon, Seattle, Washington.
2020-2025	Graduate Research/Teaching Assistant, UCLA, Los Angeles, California.

Publications

Mohamad Rida Rammal, Ruida Zhou, Suhas Diggavi. "A Statistical Approach for Measuring Data Acquisition Value," *Arxiv preprint. Arxiv:2409.00284, 2024*

Mohamad Rida Rammal, Alessandro Achille, Aditya Golatkar, Suhas Diggavi, Stefano Soatto. "On Leave-One-Out Conditional Mutual Information For Generalization," in *Neural Information Processing Systems (NeurIPS), 2022.*

Mohamad Rida Rammal, Suhas Diggavi, Ashutosh Sabharwal. "Coded Estimation: Design of Backscatter Array Codes for 3D Orientation Estimation," in *IEEE Transactions on Wireless Communications, vol. 22, no. 9, pp. 5844-5854, Sept. 2023, doi: 10.1109/TWC.2023.3237617.*

Mohamad Rida Rammal, Suhas Diggavi, Ashutosh Sabharwal. "3D Orientation Estimation With Configurable Backscatter Arrays," in *IEEE International Symposium On Information Theory (ISIT), 2022.*

CHAPTER 1

Introduction

Neural networks, and in particular language models, have proliferated in the recent years. These models have entered the mainstream due to their excellent performance on a diverse range of machine learning tasks including classification, natural language processing, speech recognition, computer vision, data generation, and forecasting. Despite this popularity, little is understood about the inner workings of deep learning models due to their opaque, black box nature. We investigate two important problems in machine learning from an information-theoretic perspective: (1) the generalization of deep learning models and (2) the value of data acquisition for large language models. We will discuss these problems in more detail below.

Information theory is the mathematical study of the quantification, processing, and transmission of information. It has its strongest ties to communication theory, where it addresses two fundamental questions: What is the optimal rate of data compression, and what is the optimal transmission rate for communication? However, information theory intersects with many other fields, not least of which are computer science and machine learning. For example, information theory plays a crucial role in areas such as data representation (e.g., variational autoencoders and generative models), model evaluation (e.g., Kullback-Leibler divergence), and feature selection (e.g., mutual information). Given this close relationship, we chose information theory as the primary language with which we investigate the problems of generalization and data valuation.

Despite being over parameterized and capable of learning any reasonably sized dataset with perfect accuracy, deep learning models still provide state-of-the-art results on a vast array

of machine learning tasks. In classical machine learning, generalization is often understood through the bias-variance tradeoff: overparameterized (complex) models tend to fit training data better but typically at the expense of worse generalization to unseen data. Conventional machine learning wisdom would indicate that deep learning models should have difficulty generalizing to unseen data, but they defy this expectation. We propose a novel generalization measure based on leave-one-out conditional mutual information. This measure is interpretable, is computationally more tractable than existing alternatives, and most importantly improves when the learning algorithm is assumed to be interpolating i.e. achieves 0 training error on the training set.

Additionally, the rapid proliferation of large language models has raised concerns over the unauthorized use of copyrighted material in training data. Large language models are trained on vast amounts of text, which can sometimes include proprietary and copyrighted data [25]. As data owners restrict access to certain datasets and offer licenses for their use, a natural question to answer is the following: How can a model owner assess the value of a dataset they are considering acquiring? We consider a hypothetical scenario involving two individuals: Alice and Bob. Alice owns a language model, while Bob possesses a dataset that Alice may acquire (or purchase) from him. Our goal is to assess the value of this dataset to Alice within the context of this transaction, which we refer to as the "acquisition value". We argue that data is less valuable if the model could have plausibly generated it on its own. Starting from that intuition, we introduce a novel data valuation function inspired by distribution testing. We demonstrate that our proposed function is computationally efficient and possesses several desirable theoretical properties.

Finally, we apply information theory to the study of 3D orientation estimation in a wireless communication setting. Orientation plays a critical role in many fields, including robotics and aerospace. We address the problem of estimating the orientation of symmetric objects equipped with configurable backscatter tags. These tags reflect incoming signals and can be configured to have two distinct reflection coefficients. By analyzing the reflected signals, we aim to determine the 3D orientation of the object. The design of the tags' reflection

coefficients can be represented as a binary code, indicating whether and when each tag reflects. Our objective is to find the optimal codes for accurate 3D orientation estimation. To this end, we propose two design criteria and provide a mathematically rigorous analysis of the performance when the designed codes are used.

1.1 Contributions and Outline

The contributions of this dissertation are aimed at enhancing the challenges described above, with the goal of applying an information theoretic point of view for better understanding of black-box deep learning models, particularly for generalization and data valuation. In the following, we provide a detailed description of the contributions made towards addressing these challenges.

In Section 2, we define the notation used throughout this dissertation and provide the necessary background for supervised and unsupervised learning.

Information-Theoretic Generalization Bounds. In Chapter 3, we investigate generalization bounds targeting deep learning models. Our contributions are as follows:

- We introduce a novel information theoretic bound based on *leave-one-out conditional mutual information* (loo-CMI). The bound improves under assumptions of interpolation. The main intuition behind the bound is to ask the counterfactual question "If I were to remove one sample at random from the training set, how well would we be able to determine which sample was the one that was removed?"

- We show that the bound is easily interpretable and computationally efficient. Moreover, we show that bounding this quantity is enough to bound the amount of memorization in a deep network, and in turn its generalization.

- We connect our bound to several empirical quantities and other generalization theories, such as stability of the optimization algorithm and the flatness of the loss landscape at

convergence.

- We provide the function-space counterpart to the bound i.e. the bound which looks at the activations of the network as opposed to the weights. This leads to a tighter bound through the data processing inequality.
- We run experiments which show that our bounds gives non-vacuous gaurantees on real-world image classification tasks. We also study its dependency on the size of the datasets and the hyperparameter.

Data Acquisition Value Function. In Chapter 3, we study the problem of data acquisition value for generative models, and in particular language models. Our contributions are as follows:

- We introduce a novel value function: the Uniform-Marginal and Independence (UMI) value function, constructed from intuitive ideas of what makes data valuable, and satisfying desirable properties for a value function. The UMI value function disentangles the challenging task of statistical testing for high-dimensional data into two simpler problems: univariate distribution testing and random variable independence testing.
- We demonstrate that the f-divergence between the marginal and uniform distributions establishes a lower bound for the f-divergence between the data model and language model.
- Experiments are conducted to illustrate the effectiveness of the proposed UMI function in detecting data generated by the model, and its behavior on data present in the training set, and new unseen data.
- We study its relationship to training outcomes by computing model performance when the training data is selected according to the UMI function.

Code Designs for 3D Orientation Estimation In Chapter 3, we study the problem of code designs for 3D orientation estimation of an object equipped with configurable backscatter tags. Our contributions are as follows:

- We propose two novel analytic code design criteria that depend on channel knowledge.
- We also develop a lower bound on the worst-case error, which quantifies the systems performance with respect to channel parameters such as the number of antennas and the number of tags used.
- We numerically investigate the performance of our designed codes with respect to a baseline orthogonal code. In some cases, our designs provide an order of magnitude improvement in error performance.
- We provide a comprehensive numerical exploration of the systems performance, including the impact of multipath, and the robustness of the design criteria against imperfect channel knowledge. We show that our codes do not require precise channel knowledge.

We provide a conclusions and potential directions for future work in Chapter 6. Finally, we provide proofs and additional experiments in the Appendix.

Tying the various themes The common theme across these different topics is the usefulness of information theory in wide range of topics. Through applying information theory to the study of non-vacuous bounds for machine learning generalization, data acquisition value function for generative models, and 3D orientation estimation for objects exhibiting symmetries, we demonstrate that the utility of information theory and its various tools is not restricted to communication, but also stretches out to other disparate topics such as machine learning, and coding.

CHAPTER 2

Background and Preliminaries

In this chapter, we define the notation and state some basic definitions that will be used throughout the thesis. We also present the background for supervised learning Section 2.2.1, and unsupervised learning in Section some background on the definitions that will be used

2.1 Preliminaries

$\mathbb{R}^k$ and $\mathbb{C}^k$ are the sets of real-valued and complex-valued $k-$dimensional vectors, respectively. We use bold lowercase letters $\boldsymbol{x}$ to denote vectors matrices and bold upper case letters $\boldsymbol{A}$ to denote matrices. $\boldsymbol{A}^T$ and $\boldsymbol{A}^H$ denote the transpose and conjugate transpose of the matrix $\boldsymbol{A}$. $\boldsymbol{A}^{-1}$ denotes the inverse of the matrix $\boldsymbol{A}$. If $\boldsymbol{x}$ is a vector with n elements, then $\text{diag}(\boldsymbol{x})$ denotes the $n \times n$ square matrix with the elements of $\boldsymbol{x}$ on its main diagonal; $\|\cdot\|$ is the standard Euclidean norm on vectors, and $\|\cdot\|_F$ is the Frobenius norm on matrices. $\langle \cdot, \cdot \rangle$ is the dot product between two vectors.

We denote with $[n]$ the set $\{1, \ldots, n\}$. We use non-bold uppercase letters to denote random variables, and non-bold lowercase letters to denote a specific value the random variable takes. If X is a random variable, then p_X is the probability distribution of X, and $\mathbb{E}_{x \sim p_X}[x]$ denotes the expected value of X. For a pair of random variables X and Y, $p_{(X,Y)}$ and $p_X p_Y$ are the joint and product distributions, respectively. If p and q are two probability densities over $\mathbb{R}^d$ such that p is absolutely continuous with respect to q ($p \ll q$),

the Kullback-Leibler Divergence from p to q is defined as:

$$\mathrm{KL}(p \parallel q) = \int_{\mathbb{R}^d} p(x) \log \left(\frac{p(x)}{q(x)} \right). \tag{2.1}$$

The Mutual Information between two random variables X and Y, and the conditional mutual information of X and Y given a third random variable Z are respectively given by:

$$I(X;Y) = \mathrm{KL}(p_{(X,Y)} \parallel p_X p_Y). \tag{2.2}$$

$$I(X;Y \mid Z) = \mathbb{E}_{z \sim P_Z}[\underbrace{I(X;Y \mid z)}_{I(X;Y|Z=z)}]. \tag{2.3}$$

2.2 Background

2.2.1 Supervised Learning

Let $Z^n = \{Z_1, \ldots, Z_n\} \in \mathcal{Z}^n$ be a collection of n independent and identically distributed samples drawn from some unknown distribution $\mathcal{P}$. We consider the setting of supervised learning where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$; i.e., each sample is composed of a feature vector and a label. Let $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W} \subseteq \mathrm{R}^K$ be a possibly stochastic training algorithm, where $\mathcal{W}$ is a set of weights (e.g., weights in a neural network). Given a loss function $\ell : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}_+$, the empirical risk for weights w on Z^n is given by $\widehat{\mathcal{L}}(w, Z^n) = 1/n \sum_{i=1}^n \ell(w, Z_i)$, and the "true" loss is given by $\mathcal{L}(w) = \mathbb{E}_{Z' \sim \mathcal{P}}[\ell(w, Z')]$; i.e., the average loss w incurs on random samples drawn from $\mathcal{P}$.

Our goal is to find a $w^* \in \mathcal{W}$ such that $w^* = \arg\min_{w \in \mathcal{W}} \mathcal{L}(w)$. Since we do not have access to the data-generating distribution $\mathcal{P}$, computing the true loss is infeasible. However, we do have access to a set of samples Z^n, and one approach is empirical risk minimization where we find the $w \in \mathcal{W}$ that minimizes $\widehat{\mathcal{L}}(w, Z^n)$. If the difference $\mathcal{L}(w) - \widehat{\mathcal{L}}(w, Z^n)$ is small, then the true loss of an empirical risk minimizer would be nearly optimal within the considered hypothesis class. Therefore, it is of great interest to study the generalization gap

which we formally define as:

$$\text{gen}(\mathcal{A}) = \left| \mathbb{E}_{\mathcal{A}, z^n \sim \mathcal{P}^n} \left[\mathcal{L}(\mathcal{A}(z^n)) - \widehat{\mathcal{L}}(\mathcal{A}(z^n), z^n) \right] \right|. \tag{2.4}$$

2.2.2 Language Models

Language models are models of the natural language. They take as input a sequence of tokens $x_{1:n} = (x_1, \ldots, x_n)$ drawn from some vocabulary $\mathcal{V}$. The models then output a probability distribution over the next token in the sequence $p_{\boldsymbol{w}}(\cdot \mid x_{1:n})$, where the probability distribution is parameterized by the weights of the model $\boldsymbol{w}$. State-of-the-art language models typically have a maximum context length L, in which case the probability of the current tokens depends only on the previous L tokens i.e. if $n > L$, then $p_{\boldsymbol{w}}(\cdot \mid x_{1:n}) = p_{\boldsymbol{w}}(\cdot \mid x_{n-L+1:n})$. A typical value for L is somewhere between 512 and 4096 tokens, but can be as large as $100,000$ or more.

Language models are trained on unlabeled data (unsupervised learning). Let $X^n = \{X_1, \ldots, X_n\} \in \mathcal{Z}^n$ be a collection of n independent and identically distributed samples drawn from some unknown distribution $\mathcal{Q}$, where $\mathcal{Q}$ is a probability distribution over $\cup_{n \in \mathbb{N}} V^n$, the set of finite-length sequences of $\mathcal{V}$. Each sample Z_i is a sequence of tokens of finite size n_i. Language models are trained by finding the set of parameters which minimize the $\boldsymbol{w}^*$ which minimize the negative log-likelihood of the training data data i.e. we find the set of weights $\boldsymbol{w}^*$ which minimize

$$L(\boldsymbol{w}) = -\sum_{i=1}^{n} \log \left(p_{\boldsymbol{w}}(x_{i,1:m_i}) \right) = -\sum_{i=1}^{n} \sum_{j=1}^{m_i} \log \left(p_{\boldsymbol{w}}(x_{ij} \mid x_{i,1:j-1}) \right). \tag{2.5}$$

CHAPTER 3

Conditional Mutual Information Bounds for Generalization

3.1 Introduction

Generalization in classical machine learning models is often understood in terms of the bias-variance trade-off: over-parameterized models fit the data better but are more likely to overfit. However, the push for ever larger deep network models, driven by their remarkable empirical generalization, has spurred a race for the development of new theories of generalization with two main objectives: (1) guiding the practitioners in designing over-parameterized architectures and training schemes that would generalize better and (2) providing provable bounds on the performance of the model after deployment.

In this context, information theoretic bounds on generalization [8, 26, 31, 47, 56, 70, 82] are particularly appealing. They capture the intuitive idea that models that memorize less about the training set will generalize better. In particular, since they deal with the amount of *information* stored in the weights rather than the *number* of weights, they are especially adaptable to over-parameterized models [2, 3, 33]. However, the bounds are often vacuous in realistic settings, require modified training algorithms, or are black box bounds that are difficult to compute and/or to relate to relevant properties of the network/training algorithm. To remedy this problem, we ask whether it is possible to develop a bound that is: (a) realistically tight for standard training algorithms used in deep learning; (b) interpretable, meaning that they can be rewritten in terms of quantities the practitioners can control during the training process and that may guide toward the design of better training algorithms, (c)

that applies out-of-the-box to standard training algorithms.

As a step in this direction, we introduce a new class of information theoretic bounsd based on *leave-one-out conditional mutual information* (loo-CMI). The bound arises from the counterfactual question "If I were to remove one sample at random from my training set and train the model from scratch on the dataset with that one sample missing, how well would I be able to infer which of these sample was the one that was removed?". We show that the conditional mutual information between training output and the identity of the dropped sample bounds the amount of memorization in the network, and thus its generalization (Theorems 3.2.1 and 3.2.2). Our bound is easy to interpret and connects to the empirical quantities, the stability of the optimization algorithm (Section 3.3.4) and the flatness of the loss landscape at convergence (Section 3.3.5). Moreover, we provide a variation of the bound in the function space in which case replace the weights of the network with the activations of the network 3.2.2). This leads to a tighter bound for networks with more parameters than training sample because many sets of parameters that the model can have can lead to identical prediction functions.

Our loo-CMI bound falls in the general class of conditional mutual information bounds, introduced by [70]. However, while standard CMI bounds iterate over all the subset of samples of size N (2^N subsets for a dataset of $2N$ samples) in order to compute the bound, our loo-CMI bound only needs to iterate over N possible samples to remove, leading to an exponential reduction in the cost to estimate the bound. Moreover, our strategy of removing a single sample is easy to relate to the stability of the training path to slight perturbation of the training set, thus giving a clear connection between generalization and the geometry of the loss landscape (Figure 3.1).Empirically, we show that our bound can be computed and is non-vacuous for state-of-the-art deep networks fine-tuned on standard image classification tasks (Table 3.1). We also study the dependency of the bound on the size of the dataset (Figure 3.2), and the hyper-parameters (Figure 3.3).

3.1.1 Problem Formulation

We restate the problem formulation here for convenience. Consider a collection of n independent and identically distributed samples drawn from some unknown distribution $\mathcal{P}$ i.e. $Z^n = \{Z_1, \ldots, Z_n\} \in \mathcal{Z}^n$, where $Z_i \sim P$. In this chapter, we consider the setting of supervised learning where each sample $Z_i = (\boldsymbol{X}_i, Y_i)$ is composed of each sample is composed of a feature vector $\boldsymbol{X}_i$ and a label Y_i.

A learning algorithm is a possibly stochastic function $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W} \subseteq \mathrm{R}^K$ which takes as input a dataset and outputs parameters $w\mathcal{W}$, the a set of all parameters (e.g., weights in a neural network). Given a loss function $\ell : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}_+$, the "true" risk is given by $\mathcal{L}(w) = \mathbb{E}_{Z' \sim \mathcal{P}} [\ell(w, Z')]$; i.e., the average loss w incurs on random samples drawn from $\mathcal{P}$. Moreover, the empirical risk for weights w on Z^n is the average loss incurred on the training dataset $\widehat{\mathcal{L}}(w, Z^n) = 1/n \sum_{i=1}^n \ell(w, Z_i)$.

The goal of the learning algorithm is to find parameters $w^* \in \mathcal{W}$ which minimize the true loss, then $w^* = \arg\min_{w \in \mathcal{W}} \mathcal{L}(w)$. However, we do not have access to the data-generating distribution $\mathcal{P}$, and hence computing the true loss is infeasible. However, we do have access to a set of samples Z^n, and one approach is to settle finding the set of parameters $w \in \mathcal{W}$ that minimizes the empirical risk $\widehat{\mathcal{L}}(w, Z^n)$. If the difference $\mathcal{L}(w) - \widehat{\mathcal{L}}(w, Z^n)$ is small, then the true loss of an empirical risk minimizer would be nearly optimal within the set of all parameters $\mathcal{W}$. The difference between the true risk and the empirical risk for $w \in \mathcal{W}$ is called the generalization gap, and is formally defined as:

$$\mathrm{gen}(\mathcal{A}) = \left| \mathbb{E}_{\mathcal{A}, z^n \sim \mathcal{P}^n} \left[\mathcal{L}(\mathcal{A}(z^n)) - \widehat{\mathcal{L}}(\mathcal{A}(z^n), z^n) \right] \right|. \tag{3.1}$$

This chapter studies upper bounds on the generalization gap $\mathrm{gen}(\mathcal{A})$. We leverage information-theoretic quantities for this purpose, specifically the mutual information. The mutual information between two random variables X and Y given by the Kullback-Leibler divergence

between the joint distribution of X and Y, and the product distribution of X and Y:

$$I(X;Y) = \mathrm{KL}\left(p_{X,Y} \,||\, p_X p_Y\right).$$

Moreover, the conditional mutual information between X and Y conditioned on a third variables Z is given by:

$$I(X;Y \mid Z) = \mathbb{E}_{z\sim P_Z}[\underbrace{I(X;Y \mid z)}_{I(X;Y|Z=z)}]. \tag{3.2}$$

We now give an overview of prior conditional mutual information bounds present in the literature.

3.1.2 Conditional Mutual Information Bounds

We start by recalling the main results on Conditional Mutual Information (CMI) bounds that we return to later. Let $\tilde{Z}^{2n} \in \mathcal{Z}^{n\times 2}$ be a dataset with $2n$ samples drawn from a distribution $\mathcal{P}$, grouped into n pairs. Let $S \sim \mathcal{S} = \text{Uniform}\left(\{0,1\}^n\right)$ a uniform random binary vector of size n. S selects one sample from each pair in $\tilde{Z}^{2n}$ to form $\tilde{Z}^n_S \in \mathcal{Z}^n$ given by $(\tilde{Z}^n_S)_i = \tilde{Z}^{2n}_{S_i+1}$. In this setting, the bound of Steinke and Zakynthinou [70] arises from asking the following question : "if a model is trained with a subset of samples chosen through the random binary indexing vector S, how much information does the output of the algorithm provide about S?" Here, the Shannon Mutual Information is useful because it is a measure of the amount of 'information' obtained about one random variable after observing another random variable.

Theorem 3.1.1 (Theorem 2, [70])
Let $\ell : \mathcal{W} \times \mathcal{Z} \to [0,1]$ be a bounded loss function. Let $\mathcal{A}$ be a possibly stochastic training algorithm. Let $W = \mathcal{A}(\tilde{Z}^n_S)$ be the output of $\mathcal{A}$ given the dataset $\tilde{Z}^n_S$, then

$$\text{gen}(\mathcal{A}) \leq \sqrt{\frac{2}{n} I(W; S \mid \tilde{Z}^{2n})}. \tag{3.3}$$

Haghifam et al. [26] improved Theorem 3.1.1 by moving the expectation over $\tilde{Z}^{2n}$ outside the square root, and by measuring the information the output of the algorithm provides on random *subsets* of the indexing vector S. Harutyunyan et al. [31] further improved the bound by moving the expectation over the random subsets outside the square root.

Theorem 3.1.2 (Theorem 2.6, [31])
Let $\ell : \mathcal{W} \times \mathcal{Z} \to [0,1]$ be a bounded loss function. Let $m \in [n]$ and let $V \sim \mathcal{V}$ be a random subset of $[n]$ of size m. Let $W = \mathcal{A}(\tilde{Z}_S)$ be the output of $\mathcal{A}$ given the dataset $\tilde{Z}_S$, then

$$\text{gen}(\mathcal{A}) \leq \mathbb{E}_{\tilde{z}^{2n}\sim\mathcal{P}^{2n}, v\sim\mathcal{V}} \sqrt{\frac{2}{m} I(W; S_v \mid \tilde{z}^{2n})}. \tag{3.4}$$

One appealing aspect of the CMI bounds is that they work for any training algorithm and any model, including over-parameterized networks trained with Stochastic Gradient Descent

(SGD). However, computing the bounds requires iterating over all possible 2^N values of S. Moreover, the bound is difficult to interpret since the value of $W = \mathcal{A}(\tilde{Z}_S^n)$ for different values of S can vary significantly and in unpredictable ways for large non-convex models. Taking inspiration from leave-one-out cross validation, we propose different CMI bound which avoids both problems by removing a single sample from the dataset. As we shall see, it allows for interpretable expressions, faster computation, and a connection to notions of stability.

3.2 Leave-One-Out Conditional Mutual Information

Let $Z^n = \{Z_1, Z_2, \ldots, Z_n\} \in \mathcal{Z}^n$ be a dataset of n i.i.d. samples drawn from $\mathcal{P}$. Let $U \sim \mathcal{U} = \text{Uniform}([n])$ be uniform random variable taking values over the indices of the samples in Z. U removes a single sample from Z^n to form $Z^{n-1}_{-U} = Z^n \setminus Z_U$, the dataset without the U^{th} sample. Let $W = \mathcal{A}(Z^{n-1}_{-U})$ be the output of the algorithm, then we define the pointwise leave-one-out conditional mutual information as:

$$\text{loo-CMI}(\mathcal{A}, z^n) = I(W; U \mid z^n). \tag{3.5}$$

In this setting, loo-CMI$(\mathcal{A}, z^n)$ measures the amount of information that the output of the algorithm W reveals about U, the index (identity) of the left-out sample. As it is the case for the conditional mutual information terms in Theorems 3.1.1 and 3.1.2, loo-CMI is bounded. Specifically, loo-CMI is upper bounded by the entropy of U, $H(U) = \log(n)$. Hence, it does not suffer from the same issues as information stability bounds [56, 82]. Moreover, the output of an algorithm is not significantly affected by the inclusion or exclusion of one sample when the input consists of thousands of other samples, so we expect loo-CMI to be small for large values of n. We use (3.5) to derive a bound on the generalization of error in expectation (Theorem 3.2.1).

3.2.1 Generalization Bounds Based on loo-CMI

loo-CMI is deeply intertwined with leave-one-out cross validation, so before we derive bounds on the generalization through loo-CMI, it is helpful to first obtain a probabilistic upper bound on the leave-one-out cross validation error (loo-cv). loo-cv measures the difference in loss between the samples which the algorithm trained on, and the sample that was left out. We begin with a precise definition of loo-cv and provide a probabilistic bound on it.

$$\text{loo-cv}(w, z^n, u) = \frac{1}{n-1} \sum_{i \neq u} \ell(w, z_i) - \ell(w, z_u). \tag{3.6}$$

Lemma 3.2.1

Let $\ell : \mathcal{W} \times \mathcal{Z} \to [0,1]$ be a bounded loss function, then for all $t > 0$, $w \in \mathcal{W}$, and $z^n \in \mathcal{Z}$,

$$\mathbb{E}_{u\sim\mathcal{U}}\left[\exp\left(t \cdot (\text{loo-cv}(w, z^n, u))\right)\right] \leq \exp\left(\frac{t^2 c_n^2}{8}\right), \qquad c_n = \frac{n}{n-1}. \tag{3.7}$$

To derive the generalization bounds included in this paper, we make heavy use of Lemma 3.2.1. We begin with a generalization bound in expectation.

Theorem 3.2.1

Let $\mathcal{A} : \mathcal{Z}^{n-1} \to \mathcal{W}$ be a training algorithm. Let $\ell : \mathcal{W} \times \mathcal{Z} \to [0,1]$ be a bounded loss function, then

$$\text{gen}(\mathcal{A}) \leq \frac{c_n}{\sqrt{2}} \mathbb{E}_{z^n\sim\mathcal{P}^n} \sqrt{\text{loo-CMI}(\mathcal{A}, z^n)}, \tag{3.8}$$

where c_n is given in (A.1). The bound in (3.8) guarantees a good generalization error when loo-CMI$(\mathcal{A}, z^n)$ is small; i.e., when the output of the algorithm W tells us little about the identity of the sample that was removed from the training dataset. If a parametric learning algorithm ends up memorizing the samples in the training set, the generalization gap could be large, and so would loo-CMI as we can determine U by checking which sample was not memorized by the algorithm. On the other extreme, a learning algorithm which outputs a constant parameter regardless of the training set does not have a generalization gap, and in this case loo-CMI would equal 0 as W provides no information about U. We note that bound of Theorem 3.2.1 most closely resembles the bound of Theorem 3.1.2 when $m = 1$. The latter bound is given by:

$$\text{gen}(\mathcal{A}) \leq \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\tilde{z}^{2n}\sim\mathcal{P}^{2n}} \sqrt{2I(W; S_i \mid \tilde{z}^{2n})}. \tag{3.9}$$

Recalling the setting of Section 3.1.2, (3.9) is computed by first going through every possible value of $S \in \{0,1\}^n$. For every value of S and for each $i \in [n]$, one fixes $S = (S_1, S_2, \ldots, S_n)$ excluding S_i which varies uniformly over $\{0,1\}$. In other words, the i^{th} component of the training set $\tilde{Z}_S^n$ is equally likely to be either $\tilde{Z}_{i,1}$ or $\tilde{Z}_{i,2}$. The right-hand-side of (3.9) then

bounds the generalization using the information that the weights contain about S_i. In other words, the question asked is: "can the output of the algorithm help us determine which one of $\tilde{z}_{i,1}$ or $\tilde{z}_{i,2}$ was present in the training set?". On the other hand, we ask a different question, "can the output of the algorithm help us determine the index of the sample that was removed?". This subtle change leads to an exponential reduction in the cost of computing the since we only need to iterate over the values of $U \in [n]$, as opposed to S which can take 2^n possible values.

3.2.2 Extension To The Function Space

The previous bounds used the output of the algorithm, the weights, to bound the generalization error. A different approach studied in [31] is to assume that given a training dataset and a test sample, the algorithm outputs a prediction on the test sample. In other words, we assume the algorithm is a possibly stochastic function $h : \mathcal{Z}^{n-1} \times \mathcal{X} \longrightarrow \mathcal{R}$ which takes a training set z, a new unlabeled sample x_{new}, and outputs a possibly stochastic prediction $h(z, x_{\text{new}})$ on the new sample. The set of predictions $\mathcal{R}$ can be different than the set of labels $\mathcal{Y}$ (e.g., class probabilities in multi-class classification tasks). Unlike bounds with respect to the weights, this approach is applicable to both parametric algorithms (e.g. neural networks) and non-parametric algorithms (e.g. k-nearest neighbors). In the former case, the output of the algorithm $\mathcal{A}$ specifies a prediction function $g \in \{g_w : \mathcal{X} \to \mathcal{R} \mid w \in \mathcal{W}\}$ from a class of functions parameterized by the weights i.e. $h(z, x_{\text{new}}) = g_{\mathcal{A}(z)}(x_{\text{new}})$.

We redefine the loss to be a non-negative function $\ell : \mathcal{R} \times \mathcal{Y} \to \mathbb{R}_+$ that measures the distance between the predictions and the ground-truth labels. Given ℓ and a training set Z^n, the true loss of an algorithm h is given by $\mathcal{L}(h, Z^n) = \mathbb{E}_{(x',y')\sim\mathcal{P}}\left[\ell(h(Z^n, x'), y')\right]$, and the empirical loss is given by $\widehat{\mathcal{L}}(h, Z^n) = 1/n \sum_{i=1}^{n} \ell(h(Z^n, X_i), Y_i)$. The generalization gap of h is then defined as:

$$\text{gen}(h) = \left| \mathbb{E}_{h, z^n \sim \mathcal{P}^n} \left[\widehat{\mathcal{L}}(h, z^n) - \mathcal{L}(h, z^n) \right] \right|. \tag{3.10}$$

Recalling the setup of Section 3.2, we define the pointwise functional leave-one-out conditional

mutual information floo-CMI as:

$$\text{floo-CMI}(h, z^n) = I\left(h\left(z_{-U}^{n-1}, x^n\right); U | z^n\right). \tag{3.11}$$

While loo-CMI measures the reduction in uncertainty about U after having known the weights, floo-CMI measures the reduction in uncertainty after having known the predictions made on *whole* dataset (including the prediction on the removed sample). We derive a bound with respect to floo-CMI(h, z^n) using a proof that closely resembles the proof of Theorem 3.2.1.

Theorem 3.2.2

Let $\ell : \mathcal{R} \times \mathcal{Y} \to [0, 1]$ be a bounded loss function. Let U be a uniform random variable over $[n]$, then

$$\text{gen}(h) \leq \frac{c_n}{\sqrt{2}} \mathbb{E}_{z^n \sim \mathcal{P}^n} \sqrt{\text{floo-CMI}(h, z^n)}, \tag{3.12}$$

where c_n is given in (A.1). As pointed out in [31], for a parametric algorithm $\mathcal{A}$ with prediction function h, U—$\mathcal{A}(z_{-U}^{n-1})$—$h(z_{-U}^{n-1}, X)$ is a Markov chain. By the data-processing inequality, we have that floo-CMI$(h, z^n) \leq$ loo-CMI$(\mathcal{A}, z^n)$. Since the bounds of Theorems 3.2.1 and 3.2.2 differ only through loo-CMI and floo-CMI, the bound of Theorem 3.2.2 is sharper.

3.3 Computing The Bounds

3.3.1 Computable Upper Bounds to loo-CMI and floo-CMI

To evaluate and interpret the bounds for a parametric algorithm $\mathcal{A}$ with prediction function h, we require a computable closed-form expressions for loo-CMI and floo-CMI. However, obtaining a closed form expression is a difficult task in most cases. To see the reason behind this difficulty, note that loo-CMI can be written as:

$$\text{loo-CMI}(\mathcal{A}, z^n) = I(W; U \mid z^n) = \mathbb{E}_{u \sim \mathcal{U}}\left[\text{KL}(p_{W \mid z^n, u} \parallel p_{W \mid z^n})\right], \tag{3.13}$$

where $p_{W \mid z^n} = \mathbb{E}_{u \sim \mathcal{U}} \left[p_{W \mid z^n, u} \right]$ is a mixture distribution. Since one of the terms in the Kullback-Leibler divergence of (3.13) is a mixture distribution, it is unlikely to obtain a closed form expression for the conditional mutual information for most commonly encountered continuous distributions (e.g., Gaussian distributions [18]). To find a computable and interpretable bound on the generalization error, we propose upper bounds on the conditional mutual information which can be interpreted and evaluated.

Theorem 3.3.1

Let $\mathcal{A}$ be a parametric algorithm with prediction function h. Let U, Z^n, W be defined as before. Let U' be an identical independent copy of U and $W' = \mathcal{A}(z_{U'}^{n-1})$, then

$$\text{loo-CMI}(\mathcal{A}, z^n) \leq -\mathbb{E}_{u \sim \mathcal{U}} \left[\ln \mathbb{E}_{u' \sim \mathcal{U}} \left[\exp \left(-\text{KL}(p_{W \mid z^n, u} \parallel p_{W' \mid z^n, u'}) \right) \right] \right], \tag{3.14}$$

$$\text{floo-CMI}(h, z^n) \leq -\mathbb{E}_{u \sim \mathcal{U}} \left[\ln \mathbb{E}_{u' \sim \mathcal{U}} \left[\exp \left(-\text{KL}(p_{h(z_{-u}^{n-1}, x)} \parallel p_{h(z_{-u'}^{n-1}, x)}) \right) \right] \right]. \tag{3.15}$$

One can alternatively upper bound loo-CMI by $\mathbb{E}_{u \sim \mathcal{U}, u' \sim \mathcal{U}'} \left[\text{KL}(p_{w \mid u, z^n} \parallel p_{w \mid u', z^n}) \right]$ through the convexity of the Kullback-Leibler divergence (similarly for floo-CMI). However, since the function $(-\ln)$ is strictly convex, it is easy to show that the bound of Theorem 3.3.1 is strictly tighter through Jensen's inequality. Moreover, we opt to use this bound as it allows for interpretable and computable expressions (Corollary 3.3.1).

3.3.2 Geometry Aware Synthetic Randomization

Theorems 3.2.1 and 3.2.2 are valid in theory, but the outputs of algorithms used in practice are deterministic given a fixed input, and so we do not have a distribution of weights or predictions. This means that (3.5) and (3.11) are degenerate. Even if we vary the random seed for neural networks trained with SGD, and run the algorithm for each dataset and random seed, we obtain a set of discrete distributions with unequal supports. Hence, the right-hand side of (A.45) would be infinite. We alleviate this issue by taking measures similar to the ones made in [26, 30, 48], and also previously in related contexts in [2, 3, 20]. In particular, we add noise to the output of a deterministic algorithm, and use the now stochastic algorithm to

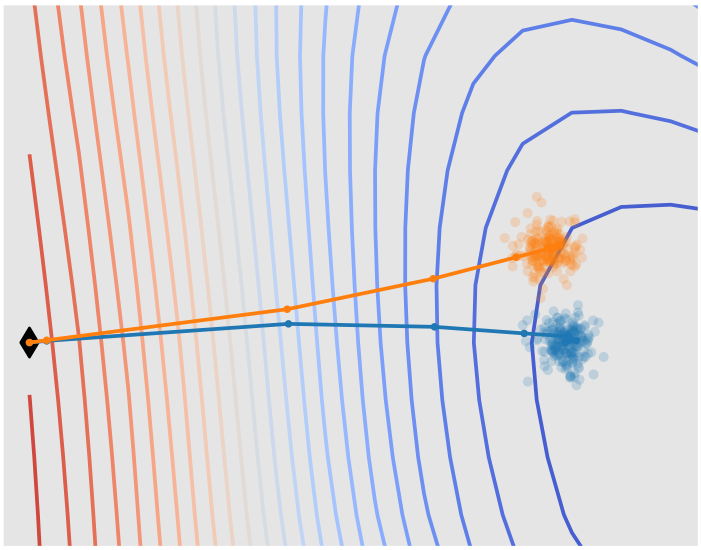

Figure 3.1: **Visualization of the quantity involved in the bound.** We plot a section of the loss landscape (contour lines) of a ResNet-18 trained on MIT-67 using the technique of [43]. In orange and blue we show the projection of training paths obtained by removing two different samples from the dataset. Equation (3.17) bounds the test error as a function of the distance between the final points of the two paths, normalized by amount of noise added (displayed by the cloud around the final weights). Note that if the loss landscape is flat we can add more noise, therefore obtaining a better bound (Section 3.3.5). If the optimization algorithm is stable, the two paths will remain close at all times, which will also improve the final bound (Section 3.3.4).

bound the information in the weights and predictions. We avoid adding isotropic Gaussian noise for the weight-based bounds, as changing the values of some weights may have little effect on the loss compared to others. Therefore, we add noise while taking into account the geometry of the loss landscape.

Specifically, let $\mathcal{A}$ be deterministic algorithm, and let $\mathcal{A}_\Sigma(z_{-U}^{n-1}) = \mathcal{A}(z_{-U}^{n-1}) + N$, where $N \sim \mathcal{N}(\mathbf{0}, \Sigma)$. A choice of Σ that would incorporate the notion that weights have a varying effect on the loss is $\Sigma = \mathrm{diag}(\alpha_1, \ldots, \alpha_K)$ with $\alpha_k > 0$ for all k and α_i not necessarily equal to α_j for $i \neq j$. Similarly, we can add noise to predictions made by a deterministic algorithm as $h_\sigma(z_{-U}^{n-1}, x) = h(z_{-U}^{n-1}, x) + M$, where $M \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$. The generalization bounds derived for $\mathcal{A}_\Sigma$ or h_σ are not directly applicable to $\mathcal{A}$ and h, but if certain Lipschitz continuity or convexity assumptions hold for the loss function ℓ, it is possible to derive bounds for deterministic algorithms from the bounds of their noisy versions (Theorem 3.3.2 and section A.1.6). We

begin by applying Theorem 3.3.1 to get interpretable expressions for the conditional mutual information terms.

Corollary 3.3.1

Let $\mathcal{A}$ be a deterministic algorithm with prediction function h. Let $w_{-i} = \mathcal{A}(z_{-i}^{n-1})$ and $h_{-i} = h(z_{-i}^{n-1}, x^n)$ be the output and predictions respectively when i^{th} sample is removed from the training set. Using Theorem 3.3.1, we obtain

$$\text{loo-CMI}(\mathcal{A}_\Sigma, z^n) \leq \ln(n) - \frac{1}{n}\sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} e^{-\frac{1}{2}(w_{-i}-w_{-j})^T\Sigma^{-1}(w_{-i}-w_{-j})}\right), \tag{3.16}$$

$$\text{floo-CMI}(h_\sigma, z^n)) \leq \ln(n) - \frac{1}{n}\sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} e^{-\frac{1}{2}\|h_{-i}-h_{-j}\|^2/\sigma^2}\right). \tag{3.17}$$

3.3.3 Connections to Stability and Bounds For Deterministic Algorithms

It is easy to see the connection between classical definitions of stability and the right-hand side of (3.16), (3.17). After all, to compute the right-hand side of (3.16) and (3.17), one only needs to know how the weights and predictions of the algorithm change when two datasets differ by one sample. Based on the observation that components of the weight might have differing degrees of effect on the loss, and the right-hand side of (3.16), we introduce the idea of "relative" weight stability. Moreover, we also use classical definitions of functional stability, and here we find it useful and intuitive to define two notions of stability: one with respect to the samples that the two datasets share, and one with respect to any other sample. The definitions are as follows.

Definition 3.3.1

Let $z^{n-1}, \hat{z}^{n-1} \in \mathcal{Z}^n$ be datasets such that z^{n-1} and $\hat{z}^{n-1}$ differ by at most one sample. Letting $a = \mathcal{A}(z^{n-1}) - \mathcal{A}(\hat{z}^{n-1})$, then we say that a deterministic algorithm $\mathcal{A}$ has ϵ weight stability relative to positive semi-definite matrix Σ if $a^T \Sigma^{-1} a \leq \epsilon^2$.

Definition 3.3.2

Let $z^{n-1}, \hat{z}^{n-1} \in \mathcal{Z}^n$ be datasets such that z^{n-1} and $\hat{z}^{n-1}$ differ by at most one sample, and without loss of generality, assume they differ at the first sample i.e. $z_1 \neq \hat{z}_1$ and $z_k = \hat{z}_k$ for all $k \neq 1$. Let h be prediction function $h : \mathcal{Z}^{n-1} \times \mathcal{X} \to \mathbb{R}^d$, then we say that h has β-train stability if $\|h(z^{n-1}, z_k) - h(\hat{z}^{n-1}, z_k)\|^2 \leq \beta^2$ for all $k \neq 1$. Moreover, we say that β_1-test stability if $\|h(z^{n-1}, x') - h(\hat{z}^{n-1}, x')\|^2 \leq \beta_1^2$ for all $x' \in \mathcal{X}$.

Given bounds on the noisy version of a deterministic algorithm, one can derive bounds on the deterministic algorithm itself by making certain Lipschitz continuity assumptions on the loss function. Moreover, we show that given relative weight stability and functional stability of deterministic algorithm, we can add an optimal amount of noise to find a bound on the deterministic algorithm. The following technique is used by [31, 48] for the case of isotropic noise. We generalize this result for arbitrary values of the noise covariance.

Theorem 3.3.2

Let $\mathcal{A} : \mathcal{Z}^{n-1} \to \mathcal{W}$ be a deterministic algorithm, and let $\ell : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}_+$ be the loss that a set of weights incur on a sample. If $\ell(w, \cdot)$ is L-Lipschitz in the weights, then if $\mathcal{A}$ has ϵ-weight stability relative to a positive semi-definite Σ, then $\text{gen}(\mathcal{A}) \leq \sqrt{4c_n \epsilon L \sqrt{\text{tr}(\Sigma)}}$.

Theorem 3.3.3

Let $h : \mathcal{Z}^n \times \mathcal{X} \to \mathbb{R}^d$ be a deterministic prediction function. Assume that h has β-train stability and β_1-test stability. Assume the loss function $\ell : \mathcal{R} \times \mathcal{Y}$ is L-Lipschitz continuity in the first coordinate, then $\text{gen}(\mathcal{A}) \leq \sqrt{4c_n L \sqrt{nd(n\beta^2 + 2\beta_1^2)}}$.

3.3.4 Bounds For Stochastic Gradient Descent

As we have seen, the quality of our information bound depends on the stability of the optimization algorithm. Stochastic Gradient Descent is the most commonly used method in machine learning, and it is therefore interesting to study the stability of SGD and how it translates into generalization from an information theoretic point of view. Assuming the gradient updates are γ-bounded for some $\gamma > 0$ (see appendix for definition), then one can bound $\|w_{-i} - w_{-j}\|$ with respect to the number of iterations of SGD [28], and obtain a bound on the generalization through the right-hand-side of (3.16). In particular, we derive the following bound which show how training for a shorter time T and having more bounded updates γ both contribute to generalization.

Lemma 3.3.1

Let the gradient update rule be γ-bounded. Suppose we run SGD for T steps, and we use the same starting value for the weights, then $\text{gen}(\mathcal{A}_{\sigma^2 I}) \leq \sqrt{\frac{c_n^2 T^2 \gamma}{\sigma^2}}$.

3.3.5 Local interpretation of the bound

The bound in Corollary 3.3.1 depends on non-local quantities as it requires retraining the model a linear number of times on subsets of the data. Ideally, one wants to bound the generalization without having to retrain. We now provide a qualitative approximation of the

bound using local quantities, in order to connect the bound with the geometry of the loss function. Assume that the training algorithm minimizes the loss $\mathcal{A}(z^n) = \arg\min_{w\in\mathcal{W}} \ell(z^n, w)$, and let $w^* = \mathcal{A}(z^n)$ be the minimum to which the algorithm converges, and similarly let $w^*_{-i} = \mathcal{A}(z^{n-1}_{-i})$. Using influence functions [41], we can approximate:

$$w^*_{-i} - w^* \approx \frac{1}{n} H^{-1}_{w^*} \nabla_w \ell(z_i, w^*) = g_i, \tag{3.18}$$

where H_w is the hessian of the loss, $\nabla_w \ell(z_i, w^*)$ is the gradient of the loss on z_i for w^*, and we have introduced the per-sample gradient at convergence rescaled by the hessian $g_i = \frac{1}{n} H^{-1}_{w^*} \nabla_w \ell(z_i, w^*)$.

Using this notation we can approximate:

$$\text{loo-CMI}(\mathcal{A}_\Sigma, z) \leq \ln(n) - \frac{1}{n}\sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} e^{-\frac{1}{2}(w_{-i}-w_{-j})^T\Sigma^{-1}(w_{-i}-w_{-j})}\right), \tag{3.19}$$

$$\approx \ln(n) - \frac{1}{n}\sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} e^{-\frac{1}{2}(g_i-g_j)^T\Sigma^{-1}(g_i-g_j)}\right), \tag{3.20}$$

$$\overset{(*)}{=} \ln(n) - \frac{1}{n}\sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} e^{-\frac{1}{2}(g_i-g_j)^T H_{w^*}(g_i-g_j)}\right), \tag{3.21}$$

where in $(*)$ we have assumed $\Sigma^{-1} = H_{w^*}$, which is the optimal variance of the noise (Section 3.3.2).

From (3.21) we see that converging to a flat minimum, that is, having a small norm of the hessian H_{w^*}, is expected to correlate to better generalization. This is in accordance with several empirical results connecting convergence to flat minima to better generalization [11, 20, 34]. It should be noted that flatness by itself cannot explain generalization, since we can reparameterize the network to have identical predictions (and hence generalization) but arbitrarily large hessian [17]. Indeed, in (3.21) we see that the role of the hessian is mediated by the similarity of the (re-scaled) per-sample gradients at convergence, which can change under reparameterization. In particular, studying flatness by itself is not enough.

3.4 Related Work

Over the past several years, there has been a significant line of research in using information theory both to interpret the behavior of DNNs [2, 3, 64] and to derive generalization bounds [8, 26, 31, 47, 56, 70, 82], with the earliest works on this line from [4, 56, 82]. Many of them develop information stability bounds which involve the mutual information between the output of the algorithm and the samples [8, 47, 56, 82] which could degenerate, *e.g.,* if the data is continuous. This was observed in [70], which then proposed a new bound based on the conditional mutual information (see Theorem 3.1.1) which is non-degenerate. This work was further extended by [26, 31]. In particular [31] developed bounds based on prediction outputs rather than the algorithm outputs, and its combination with the conditional mutual information framework made it the state-of-the-art in terms of information-theoretic generalization bounds. The two issues identified in our work related to this is one of computability (as the [31] bound might in principle require computation exponential in the size of the dataset) and interpretability; this is the main focus of our work. In particular, our loo-CMI framework can not only be computed more efficiently, but also connects to classical leave-one-out cross validation measures used extensively in practice (see Theorem 3.2.1). Another aspect introduced in [31, 48] is the application of CMI bounds to deterministic algorithm outputs (as one can only run a finite number of runs of even a randomized algorithm on a dataset). We use this viewpoint in our work, but using a geometry-aware method (see Theorem 3.3.2), which takes into account the loss landscape in the algorithmic output space. There has also been a line of work connecting SGD to generalization including earlier works in [27] and more recently its connection with information theoretic bounds [48, 52]. We apply these ideas to the framework of loo-CMI (see Lemma A.1.7). The connection to classical notions of algorithmic stability can also be made using the loo-CMI framework (see Theorem A.1.6).

3.5 Experiments

Model and datasets. We now study the behavior of our loo-CMI and floo-CMI bounds on real-world image classification tasks. In particular, we fine-tune an off-the-shelf ResNet-18 model pretrained on ImageNet on a set of standard image-classification tasks (see also Table 3.1): Oxford Flowers [49], MIT-67 [54], Oxford Pets [51]. On all datasets we fine-tune for 10 epochs using stochastic gradient descent with learning rate $\eta = 0.05$, momentum $m = 0.99$, batch size $B = 256$, and weight decay $\lambda = 0.0005$.

Non-vacuous bounds. In Table 3.1, we compute the loo-CMI and floo-CMI bounds using (3.8) and (A.39). We show that while the former bound is vacuous, the floo-CMI bound provides non-vacuous generalization bounds on all datasets. This is significant, as obtaining non-vacuous bound for large-scale models remains a challenging problem. The failure of the bound based on loo-CMI is expected here, since loo-CMI looks directly at the information contained in the weights of the network without considering how this information is used. In large models with millions of parameters, most of the information in the weights do not significantly affect the predictions and should ideally be discarded. This is indeed done by the floo-CMI bound.

Effect of the dataset size. To show how the quality of the bound changes for different sizes of the dataset, we subsample randomly and without replacement n samples from each dataset. In Figure 3.2, we plot the resulting generalization gap alongside our generalization bounds as the size of the subsample varies (in terms of fractions of the dataset). We observe that the generalization bound becomes tighter as the size of the training set grows. This is not surprising as we expect the model to become more stable for larger datasets.

Synthetic randomization. In Section 3.3.2, we introduced artificial Gaussian noise $N(0, \Sigma)$ in order to obtain better bounds. Increasing the noise variance Σ in the loo-CMI bound, or σ in floo-CMI, improves the bound on the CMI, but also increases the test error of the model.

Dataset	Train Error	Test Error	loo-CMI	floo-CMI
MIT-67 [54]	0	0.339	9.65	0.397
Ox. Flowers [49]	0	0.254	6.92	0.435
Ox. Pets [51]	0	0.134	8.51	0.291

Table 3.1: **Generalization bounds on image-classification tasks.** We report the train and test error (fraction of wrong predictions) and our generalization bounds on several image classification dataset ($\sigma = 0.1$).

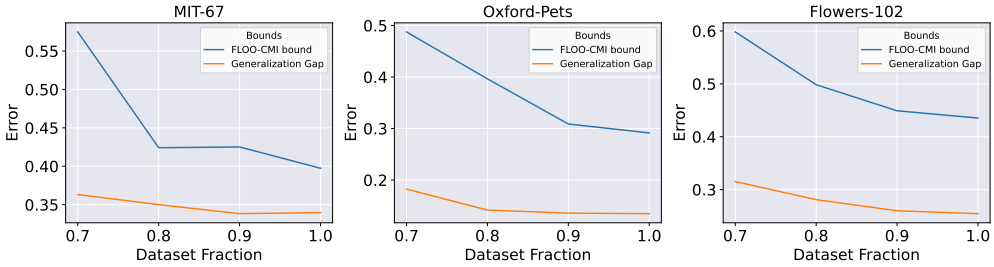

Figure 3.2: floo-CMI **bound and dataset size**. For different datasets, we plot of the floo-CMI bound as the fraction of samples used in training increases. We observe that larger datasets have better generalization bounds. This is expected as stability improves with larger datasets.

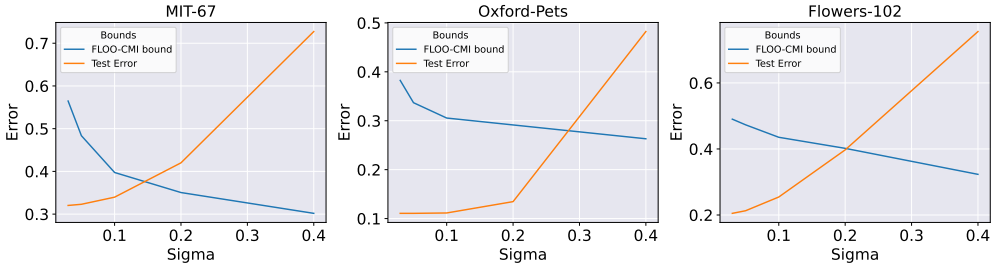

Figure 3.3: **Effect of varying σ.** Increasing the value of σ in (A.39) improves the floo-CMI bound (blue line) but also makes h_σ increasingly different from h (as shown by the increased test error, orange line), thus making bounding the generalization of the original algorithm h given h_σ more difficult. For this reason, we want to use intermediate values of σ.

Hence, we need to select a value of Σ and σ that provides a good trade-off between the two.

In Figure 3.3, we show how changing the noise variance affects each term.

3.6 Conclusion

In this chapter, we introduced a bound based on a new information-theoretic measure, *i.e.,* leave-one-out conditional mutual information and analyzed its properties. loo-CMI bounds reduce the computational costs compared to prior CMI bounds, connects to several empirical quantities and generalization theories. It leads to non-vacuous bounds for several deep learning applications in image classification. There are several open questions including, whether it is possible to (i) obtain tighter bounds using the leave-one-out framework for general learning algorithms; (ii) further reduce the computational cost of computing loo-CMI[1] and (iii) obtain deeper understanding the effect of loss landscape and dynamics of learning algorithms on learning performance using the information-theoretic lens.

[1]We point to [6], which we were made aware of at the end of the review process. In particular, this work reduces the cost of computing the leave-one-out error in the kernel regime, although it does not directly connect to the loo-CMI framework of our paper, it might be possible to combine them to make progress on this question.

CHAPTER 4

Statistical Methods for Data Acquisition Value

4.1 Introduction

In recent years, large language models (LLMs) have surged in popularity, thanks to their exceptional performance across a wide range of natural language processing tasks. However, this rapid growth has also raised concerns about the unauthorized use of copyright-protected material in developing these language models. As data owners restrict access to certain datasets and offer licenses for their use, a natural question arises: How can a model owner assess the value of a dataset they are considering acquiring?

To explore this, we consider a hypothetical scenario involving two individuals: Alice and Bob. Alice owns a language model, while Bob possesses a dataset that Alice may acquire (or purchase) from him. Our goal is to assess the value of this dataset to Alice within the context of this transaction, which we refer to as the "acquisition value". While Alice could use the dataset for various purposes—such as training or evaluation—we do not assume any specific use case at this stage. Instead, we begin with the premise that the dataset is relevant to Alice, and our objective is to determine its value to her within the transaction. It is important to note that this differs from the typical data valuation problem, which focuses on fairly allocating credit for a model's performance among the training data [23, 35, 37, 39, 80, 83].

One approach for determining acquisition value might involve training the model on the dataset (often multiple times) to assess its impact on performance. However, training large language models can be prohibitively expensive and time-consuming, and the results may vary significantly depending on the training method. A dataset might be considered valuable

with one algorithm but less so with another due to variability in outcomes and the specific task on which the model is evaluated. In this work, we propose an intuitive methodology that efficiently quantifies the acquisition value of data for language models efficiently through statistical means, rather than relying on optimization or extensive training.

The main intuition behind our approach arises from replacing the challenging question, "what makes data valuable?" with the closely related but more manageable "What data is not worth acquiring?" For a generative model, such as a large language model, there is a natural answer to the latter question: data that can be generated by the model itself. Referring back to the earlier scenario, if Bob's dataset was generated by Alice's model, we would assign a low acquisition value to the data, as Alice can generate it herself without needing to acquire it.

We posit that the acquisition value of data decreases if it can be plausibly generated by the model itself. Our primary focus is then to assess how challenging it would be for Alice to generate this dataset independently, effectively measuring its informativeness or novelty from the model's perspective. The remaining question is how to quantitatively assess this plausibility.

In this framework, we view the language model as a token predictor denoted by $p(\cdot|x_{1:n})$, where $x_{1:n} = (x_1, \ldots, x_n)$ represents the history or context of tokens used to predict the next token x_{n+1}. Here, each token is drawn from a vocabulary $\mathcal{V}$. As the owners of the language model being used for data valuation, we have access to this probability distribution $p(\cdot|x_{1:i})$. Consequently, the plausibility question revolves around assessing the statistical disparity between the model and the data. The greater this disparity, the more valuable the data. This perspective aligns with the classical framework of distribution testing, where the goal is to determine whether a given sequence of data is generated from a specified model (see [79] and references therein).

However, language models present two significant challenges: (i) the state space is massive; the alphabet size of tokens (the number of possible values tokens can take), $|\mathcal{V}|$, is typically in the tens of thousands. A typical context length L, the maximum number of tokens the model can remember at any given time, can also be somewhere in the several thousands,

necessitating methods which are computationally efficient. Moreover, (ii) to provide any performance guarantees with such a large state space, the length of data sequences needs to be enormous, often much larger than $|\mathcal{V}|^L$. This necessitates methods capable of operating effectively with much smaller data sequences, as obtaining single datapoint sequences with these lengths is unrealistic.

To address this, we develop a measure of value using Rosenblatt's transformation, which is a mapping that converts *continuous* random vectors from any distribution into uniform random variables [55]. Specifically, suppose $Y = (Y_1, \ldots, Y_d) \sim F$ is a continuous random vector distributed according to the cumulative distribution function (CDF) F, then $Z_i = F(Y_i \mid Y_{1:i-1}) \sim \mathcal{U}$ is a sequence of independent and identically distributed random variables with the standard uniform distribution over $(0, 1)$. Because Rosenblatt's transformation is only valid for continuous random variables, we first convert our discrete input tokens $x_1, \ldots, x_n$ into continuous variables $y_1, \ldots, y_n$. We then apply this transform using a continuous representation of probability distribution $p(\cdot|x_{1:i})$ output by the model.

We prove that if the sequence of tokens x_i were indeed generated from the model, then the obtained sequence of variables after transformation Z_i are independent and uniform. Using this we develop an efficient method for data value for language models by using a distribution distance function between the empirical distribution of $Z_1, \ldots, Z_n$ with the i.i.d., $\mathcal{U}$ distribution.[1]

Rosenblatt's transformation disentangles the difficult task of statistical testing for complex data (text data for language models in particular) into two simpler problems: testing for uniform distribution and marginal distribution, and independence testing. We then propose the Uniform-Marginal and Independence (UMI) value function which consists of the two aforementioned tests. The UMI value function is computationally efficient, and requires a relatively small number of tokens for accurate tests (statistical efficiency).

We show that statistical distance (f-divergence) between the marginal and uniform distributions lower bounds the statistical distance between the distribution provided by the

[1]We need to also add an independence test for this; see details in Section 4.3

language model and the true data generating distribution (see Theorem 4.3.3) . Moreover, we show that the two distances are exactly equal in the simpler univariate case (see Theorem 4.3.2).

In Section 4.4, we conduct experiments to illustrate the effectiveness of the proposed UMI function in detecting data generated by the model, and its behavior on other categories of data (training data, unseen data). We also study the relationship between the UMI function and training outcomes.

4.2 Preliminaries

4.2.1 Setup

We consider language models that take as input a sequence of tokens $x_{1:n} = (x_1, \ldots, x_n)$ drawn from a vocabulary $\mathcal{V}$. Given $x_{1:n}$, the models outputs a probability distribution over the next token in the sequence $p(\cdot \mid x_{1:n})$. Modern language models usually have a maximum context window L. This refers to the maximum amount of tokens that the model can "remember" in a single interaction. In other words, the probability distribution $p(\cdot \mid x_{1:n})$ depends on the previous L tokens only: $p(\cdot \mid x_{1:n}) = p(\cdot \mid x_{n-L+1:n})$. A typical value for L is somewhere between 512 and 4096 tokens.

We consider a dataset $\mathcal{D} = \{d_1, \ldots, d_N\}$ composed of multiple samples. Each sample or datapoint consists of a sequence of tokens of possibly variable length. Our goal is to find a measure of value for the dataset $\mathcal{D}$ as a whole assuming that our knowledge is represented through a language model, i.e., the transition probability distribution p it determines. In other words, we would like to find a value function $V : \mathcal{D} \mapsto \mathcal{R}_{\geq 0}$ which maps the dataset to a non-negative number and satisfies a number of desirable properties.

4.2.2 Properties of a Desirable Value Function

We argue that a good value function V should possess several key properties and that it should align with some of our intuitive notions about valuable data. Specifically, we are looking for a function which satisfies the following criteria:

Additivity: Given a dataset $\mathcal{D} = \{d_1, \ldots, d_N\}$, the value function V of the dataset should be additive in terms of the data points, *i.e.,* $V(\mathcal{D}) = \sum_{i=1}^{n} V(\{d_i\})$. Assuming $V(\{d\}) \geq 0$ for all d, this would immediately imply that if $\mathcal{D}_1$ and $\mathcal{D}_2$ are two datasets such that $\mathcal{D}_1 \subseteq \mathcal{D}_2$, then $V(\mathcal{D}_1) \leq V(\mathcal{D}_2)$.

Remark on Duplicates: We assume that $d_i \neq d_j$ for $i \neq j$, meaning that our dataset contains no duplicates. Duplicates are typically considered errors in language datasets, so we include a deduplication step prior to applying our value function (one can also use algorithms such as bloom filters to remove near-duplicate datapoints). This is a standard step in data filtering pipelines [44].

Baseline: We require a quantifiable criterion for determining when a data value reaches zero, establishing a baseline against which we can make comparisons. Additionally, as previously outlined, we would like this to be a statistical property rather than one based on optimization or training.

Efficient: V should be computationally feasible even for large vocabularies and context sizes. Moreover, V should prioritize sample efficiency, enabling evaluation for sequences of tokens regardless of length without the necessity for excessively lengthy datapoints.

4.2.3 Overview of the Solution

As previously mentioned, our approach stems from a shift in perspective: instead of focusing on what makes data valuable, we pivot towards identifying the data that does not carry value. In the context of a generative model, data generated internally holds no value or relevance to the model's owner. This then serves as the foundation of our approach. By initially focusing on identifying data generated by the model, we establish a starting point for developing a

value function. To identify data generated by the model, we make use Rosenblatt's probability integral transform, see (4.3).

In particular, we are given a sequence of tokens $x_1, \ldots, x_n$, where each token maps to a unique integer in the set $\{1, \ldots, |\mathcal{V}|\}$. To prepare for Rosenblatt's transformation, which requires continuous variables, we construct a sequence of continuous variables $\tilde{x}_1, \ldots, \tilde{x}_n$ using the transformation $\tilde{x}_i = x_i - u_i$, where u_i is a standard uniform random variable independent of x_i. Our language model provides a sequence of probability mass functions over the next token $p(\cdot|x_{i-L:i-1})$, one for each token in the input. We can transform these discrete probability functions provided by the model into continuous representations given by

$$\tilde{F}\left(\tilde{x} \mid x_{i-L:i-1}\right) = \sum_{j=1}^{\lfloor \tilde{x} \rfloor} p\left(j \mid x_{i-L:i-1}\right) + \left(\tilde{x} - \lfloor \tilde{x} \rfloor\right) p\left(\lfloor \tilde{x} \rfloor \mid x_{i-L:i-1}\right). \tag{4.1}$$

Finally, we evaluate the continuous y_i's at their corresponding continuous distribution functions to obtain the new sequence $z_i = \tilde{F}\left(y_i \mid x_{i-L:i-1}\right)$. We prove that if the sequence of tokens were generated by the language model, then the z_i's must necessarily be independent and identically distributed $\mathcal{U}$ random variables (see Theorem 4.3.1).

To obtain our value measure, we compare the actually obtained distribution of the z_i's to what their distribution would have been generated by the model i.e., the uniform distribution. Specifically, let G be the distribution of the z_i's, then we compute $D_f(G\|\mathcal{U})$, where $D_f(G\|\mathcal{U})$ is the f-divergence between the two distributions G and $\mathcal{U}$.[2] Moreover, since Theorem 4.3.1 guarantees both uniformity and independence of the z_i's, we employ independence testing to address edge cases where the z_i's are uniform but lack independence.

4.2.4 Alternatives

In this section, we explore two other potential candidates we have considered for our value function, namely one centered on compression and the other on identity testing. We explain

[2] f-divergence is a generalization of Kullback-Leibler divergence [13]; see Definition 4.3.1. For numerics, we use the more familiar Kullback-Leibler divergence and total variation distance.

why we chose our method over these alternatives and discuss the insights gleaned from them that steered us towards our proposed approach.

Compression: In information theory, prediction is often viewed from the lens of compression. If you can compress some data down to a small size, then it likely does not contain much new information beyond what you already knew (and vice versa). In particular, arithmetic coding is a technique which attains near-optimal compression rates when coupled with an accurate model (distribution) of the data. Large language models, coupled with arithmetic coding, have been shown to excel at compressing text data [15,74]. The compression approach provides a concrete measure of data value: compress the dataset with arithmetic coding and use the compressed size (in bits or nats) as the measure.

Let $x_{1:n} = (x_1, x_2 \ldots, x_n)$ be a token sequence generated by a distribution q over $\mathcal{V}^n$. It can be shown that any compression scheme corresponds to a distribution r over $\mathcal{V}^n$, where the number of nats used to compress $x_{1:n}$ is $1/r(x_{1:n})$. A standard result asserts that the expected compressed data size L per input token is bounded below by $\mathcal{H}(q) + \mathrm{KL}(q\|r)$, where $\mathcal{H}$ is the entropy rate and $\mathrm{KL}(q\|r)$ is the Kullback-Leibler divergence between q and r [13].

Therefore, defining the value function as $V(\mathcal{D}) = L - \mathcal{H}(q)$ ensures non-negativity, additivity, and a well-defined baseline since $V(\cdot) = 0$ when $r = q$. However, computing $\mathcal{H}(q)$, the true entropy rate of the token sequence X^n, is non-trivial for large alphabet sizes.[3] In particular, the min-max regret analysis of universal data compression shows that the estimation gap goes down as $O(|\mathcal{V}|^L/n)$ [63,81], where the token sequence is generated by an (unknown) L-order Markov chain; which means that for sources with large memory L and token alphabet size, we need an extremely long sequence to obtain a good estimate, something that is often not feasible.

Identity Testing: Identity testing, also known as goodness-of-fit, deals with the following hypothesis testing problem: Given an *explicit* description of a probability distribution p, samples from an unknown distribution q, and bound $\epsilon > 0$, distinguish with high probability

[3]Note that compressing using a language model is like compressing with some r, gives an upper bound to the unknown entropy rate.

between hypothesis $p = q$ and hypothesis $\|q - p\|_1 > \epsilon$ whenever q satisfies one of these conditions [1, 9, 16, 78]. Identity testing presents a potential method of answering the question 'Is this data valuable?' where the threshold for "valuableness" is dictated by the constant ϵ.

However, the sample complexity required to solve the identity testing problem severely diminishes the practical utility of this method, especially in the context of large language models where p can be well approximated by a high-order Markov chain. In this setting, identity testing requires datasets containing most of the states in the chain, a demand that far exceeds the feasibility of any realistic dataset. Specifically, the requisite number of samples needed to solve the identity testing problem scales as $\Omega\left(\mathcal{V}^L/\epsilon^2\right)$ [79]. Nevertheless, we leverage some insights from distribution testing to propose our own approach for measuring value.

4.3 Value function and its properties

The hardness of the identity testing problem for complicated sources, e.g., natural languages, originates from the dependency among the samples and the ever-changing next token probabilities as the context grows. To this end, we propose to use Rosenblatt's transformation [55] which decouples the dependency among tokens and transforms the changing probabilities into the fixed uniform distribution. Rosenblatt's transformation serves as an extension to the probability integral transform (PIT), originally designed for univariate i.i.d. samples. It can be viewed as iteratively applying the PIT using the chain rule in probability (see Appendix B.1.1).

Theorem 4.3.1 (Rosenblatt's transformation [55])
Let $X = (X_1, X_2, \ldots, X_d)$ be a continuous d-dimensional random vector with a joint density function

$$f(x_1, \ldots, x_d) = f_1(x_1) f_2(x_2 \mid x_1) \cdots f_d(x_d \mid x_{1:d-1}). \tag{4.2}$$

Let $F_i(\cdot|x_{1:i-1})$ be the conditional cumulative distribution function corresponding to the conditional density function $f_i(\cdot|x_{1:i-1})$. The random variables $Z_1, \ldots, Z_d$ given by Rosenblatt's

transformation

$$\begin{aligned} Z_1 &= F_1(X_1), \\ Z_i &= F_i(X_i \mid X_{1:i-1}), \quad i = 2, \ldots, d, \end{aligned} \tag{4.3}$$

are independent and identically distributed following standard uniform distribution $\mathcal{U}$*.*

Theorem 4.3.1 establishes that Rosenblatt's transformation provides an effective and unified approach for testing the plausibility of data. For example, given a random sequence $X = (X_1, \ldots, X_d)$ and a reference distribution F^{ref}, we can obtain $Z = (Z_1, \ldots, Z_d)$ by Rosenblatt's transformation of X according to F^{ref} as in Equations (4.3). X follows the reference distribution F^{ref} if and only if the random vector Z follows the uniform distribution over the d-dimensional hypercube $[0,1]^d$; otherwise, it will not. We thus have an equivalent statement for the identity hypothesis testing:

$$\begin{cases} H_0: \ X \sim F^{\text{ref}} \\ H_1: \ X \not\sim F^{\text{ref}} \end{cases} \quad \Leftrightarrow \quad \begin{cases} H_0: \ Z \sim \text{Unif}([0,1]^d) \\ H_1: \ \text{otherwise} \end{cases} \tag{4.4}$$

Note that the testing on the right-hand side can be further decomposed into two terms: first, testing the closeness, which can be measured by any f-divergence, of the averaged marginal distributions of $Z_1, \ldots, Z_d$ to standard uniform distribution $\mathcal{U}$. Second, testing the independence among $Z_1, \ldots, Z_d$. Both of these are simpler and more standard tests in statistics. Motivated by this, we propose the Uniform-Marginal and Independence (UMI) value function that encapsulates both tests.

4.3.1 The Components of the Value Function

From Discrete to Continuous. Rosenblatt's transformation in Eq (4.3) can only be applied to continuous random variables.[4] However, given that the token vocabulary and probability mass functions generated by the language model are discrete, we adapt by transforming both

[4]Continuity is required everywhere except on a set of Lebesgue measure 0, commonly referred to as almost everywhere.

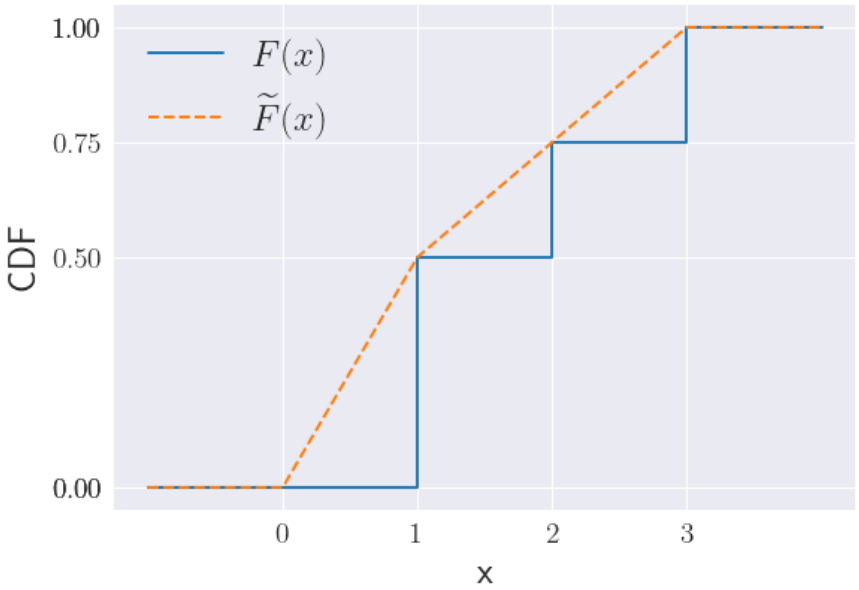

Figure 4.1: The CDF of X, which takes values in $\{1, 2, 3\}$ with probabilities $(1/2, 1/4, 1/4)$, and the interpolated CDF $\tilde{F}$ of the continuous $\tilde{X}$.

into continuous forms. Since disparate discrete outcomes are mapped to disjoint intervals on the real line, our choice of transformation will have no effect on what the value function is trying to capture (see Theorem 4.3.2).

Let X be a random variable taking values in the discrete set $[k] = \{1, \ldots, k\}$ with a probability mass function p. We construct a continuous random variable $\tilde{X}$ that takes values in the range $(0, k)$ through the following procedure: draw the discrete random variable X according to p, draw a uniform random variable U, and set $\tilde{X} = X - U$. In other words, $\tilde{X}$ will uniformly assume a value in the range $(X - 1, X)$. The probability density function of $\tilde{X}$ will be then given by $\tilde{f}(y) = p(\lceil y \rceil)$ for $0 < x \leq k$. The CDF of $\tilde{X}$, $\tilde{F}$, will be the linear interpolation of F, the CDF of the original variable X (Figure 4.1).

Comparing Against the Uniform. We employ f-divergences to quantitatively assess the disparity between the resultant marginal distribution post-Rosenblatt's transformation and the uniform distribution. f-divergences represent a diverse set of functionals which include measures such as total variation distance and Kullback-Leibler divergence. Additionally, we establish theoretical results linking the f-divergence between model and data distributions to that of the marginal and uniform distributions. We formally define f-divergences below.

Definition 4.3.1

Let P and Q be two probability distributions over $\mathbb{R}^d$ such that P is absolutely continuous with respect to Q ($P \ll Q$), and let $f : (0, \infty) \to \mathbb{R}$ be a convex function with $f(1) = 0$, then f-divergence $D_f(\cdot\|\cdot)$ is a functional that maps the pair of distributions into a non-negative number. It is given by

$$D_f(P \,||\, Q) = \mathbb{E}_Q \left[f\left(\frac{\mathrm{d}P}{\mathrm{d}Q} \right) \right], \tag{4.5}$$

where $\mathrm{d}P/\mathrm{d}Q$ is the Radon-Nikodym derivative of P with respect to Q.

For simple cases (all the cases studied in this work fall into this category), absolute continuity means $\mathrm{Supp}(P) \subset \mathrm{Supp}(Q)$, and $\mathrm{d}P/\mathrm{d}Q$ is just the likelihood ratio of the two distributions. We obtain the total variation distance when $f(x) = 1/2\,|x - 1|$ and the Kullback-Leibler divergence when $f(x) = x \log x$.

Independence Testing. Theorem 4.3.1 does not only assert a sequence of uniform random variables, but also guarantees their independence. However when dealing with tokens that were not generated by the model, there is a possibility of encountering variables with a nearly uniform marginal distribution that are nonetheless dependent. The goal of independence testing is to detect such cases.

To give an example of the tests we have used, we describe the maximum-of-t test [40]. Given a sequence of input numbers $z_1, z_2, \ldots, z_{mt}$, we divide the sequence numbers into m groups of t elements each, that is, $(z_{jt}, \ldots, z_{jt+t-1})$ for $0 \le j \le m$. We find the maximum of each group to obtain $y_1, \ldots, y_m$. We then apply a Kolmogorov-Smirnov test to the values $y_1, \ldots, y_n$ with the distribution function $F(x) = x^t$ for $0 \le x \le 1$.[5] For a sequence $Z_1, \ldots, Z_t$ of independent variables with a common distribution function $G(z)$, the probability that $\max(Z_1, \ldots, Z_t) \le z$ is the probability that $Z_i \le z$ for $1 \le i \le t$, which is just the product of the individual probabilities $G(z)G(z) \cdots G(z) = G(z)^t$ (see Appendix B.2 for a discussion on other independence tests).

[5]The Kolmogorov-Smirnov test is used to test whether a sample came from a reference one-dimensional probability distribution.

Algorithm 1 Maximum-of-t Test

Input: Sequence of numbers, $Z_1, Z_2, \ldots, Z_n$
  Group size, $t = 2, 3, \ldots$
  p-value, $p > 0$
for $i = 1$ to $\left\lfloor \frac{n}{t} \right\rfloor$ **do**
  Find $V_i = \max\{Z_{ti}, Z_{ti+1}, \ldots, Z_{ti+t-1}\}$
end for
Apply the Kolmogorov-Smirnov test to $V_1, \ldots, V_{\lfloor \frac{n}{t} \rfloor}$ with distribution $F(x) = x^t$
if the p-value $> p$ **then**
  Return True
else
  Return False
end if

4.3.2 The UMI Value Function

Given a sequence of tokens $(x_1, \ldots, x_n)$, we first calculate a sequence of probability mass functions $p(\cdot \mid x_{1:i-1})$ using our language model. We compute the continuous versions of the tokens $\tilde{x}_1, \ldots, \tilde{x}_n$ by taking $\tilde{x}_i \sim \text{Unif}(x_i - 1, x_i)$, and the continuous CDFs $\tilde{F}(\cdot|x_{1:i-1})$ corresponding to $p(\cdot|x_{1:i-1})$. Note that $x_i = \lceil \tilde{x}_i \rceil$, so we can equivalently write $\tilde{F}(\cdot|\tilde{x}_1, \ldots, \tilde{x}_n)$ without ambiguity. Given the continuous random vector $(\tilde{x}_1, \ldots, \tilde{x}_n)$ and reference CDF $\tilde{F}$, Rosenblatt's transformation can be applied as

$$\begin{aligned} z_i &= \tilde{F}(\tilde{x}_i \mid x_{1:i-1}), \\ &= \sum_{j=1}^{\lfloor \tilde{x}_i \rfloor} p(j \mid x_{1:i-1}) + (\tilde{x}_i - \lfloor \tilde{x}_i \rfloor) p(\lfloor \tilde{x}_i \rfloor \mid x_{1:i-1}), \\ &= \sum_{j=1}^{x_i} p(j \mid x_{1:i-1}) + u_i p(x_i \mid x_{1:i-1}), \end{aligned} \tag{4.6}$$

where $u_i = \tilde{x}_i - x_i$. We then conduct marginal distribution test and independence tests on $(z_1, \ldots, z_n)$. For the marginal distribution test, we first calculate the empirical distribution given by $G_n = 1/n \sum_{i=1}^{n} \mathbb{1}_{Z_i \leq z}$, where $\mathbb{1}$ is the indicator function. We can compute its interpolated version $\tilde{G}_n$ by linearly interpolating the adjacent jumps in G_n. The difference between $\tilde{G}_n$ and the reference standard uniform distribution $\mathcal{U}$ is measured via an f-divergence,

$D_f(\tilde{G}_n \| \mathcal{U})$ [53].

If the computed f-divergence $D_f(\tilde{G}_n \| \mathcal{U}) < \epsilon$ for some small constant, indicating a near-uniform marginal distribution, we run multiple independence tests to make sure our data does not fall into the edge case where the marginal is uniform, but the z_i's lack independence. If the independence tests fail, we assign a fixed value of $\alpha > 0$ to the datapoint. Both ϵ and α are user-defined hyperparameters (see Appendix B.3.3 for a discussion on how to choose them).

Let $\text{Indep}(Z)$ be the result of the independence test, returning True if the test passes and False if it fails. The Uniform-Marginal and Independence (UMI) value function can then be written as:

$$\text{UMI}(x_{1:n}) = \alpha + \mathbf{1}_{\neg \text{Indep}(Z) \vee D_f(\mathcal{U} \| \tilde{G}_n) \geq \epsilon}(D_f(\mathcal{U} \| \tilde{G}_n) - \alpha). \tag{4.7}$$

It is easy to see that the proposed UMI value function satisfies the desired properties discussed in the previous sections. In particular, our approach requires no training, relying solely on evaluating the model for the given input sequence.

4.3.3 Analysis

We relate the f-divergence between the marginal distribution of z and uniform distribution with the f-divergence between the data distribution and the language model.

4.3.3.1 IID Case

Let P and Q be two probability distribution over the finite set of tokens $\mathcal{V}$, and let F_p and $\tilde{F}_p$ (resp. F_q and $\tilde{F}_q$) be the cdf corresponding to P (Q) and its continuous counterpart, respectively. Suppose X_q is a r.v. distributed according to F_q and $\tilde{X}_q$ according to $\tilde{F}_q$. Let $Z = \tilde{F}_p(\tilde{X}_q)$ be the Rosenblatt's transformation of $\tilde{X}_q$ by a possibly mismatched CDF $\tilde{F}_p$ and let G be its CDF. Note that $G = \mathcal{U}$ if $P = Q$. We show below the equivalence of their f-divergences, which indicates the measure under the marginal distribution of Z can indeed

quantitatively capture the distance between the distributions in the token space.

Theorem 4.3.2

Let P, Q and G be as described above, then for any f-divergence, we have

$$D_f(Q||P) = D_f(G\|\mathcal{U}). \tag{4.8}$$

Remark: The empirical distribution G_n satisfies $\|G_n - G\|_\infty \leq \sqrt{\frac{\ln(2/\delta)}{2n}}$ with probability at least $1-\delta$ by Dvoretzky–Kiefer–Wolfowitz inequality [19, 46]. By the continuity of G, we can similarly obtain that $\|\tilde{G}_n - G\|_\infty \leq 2\sqrt{\frac{\ln(2/\delta)}{2n}}$ with probability at least $1-\delta$, for the interpolated $\tilde{G}_n$ used in the calculating the value.

4.3.3.2 Markov Case

The stochasticity of Language can be approximated via m-gram Markov model [60] and the approximation is more accurate for larger m. We illustrate the proposed method with measure $D_f(\tilde{G}_n\|\mathcal{U})$ gives a sufficient condition for distinguishing Markov sources.

Let P and Q be two transition kernels for m-gram Markov chains over the finite token set $\mathcal{V}$. Suppose the Markov processes determined by P and Q converge to their unique stationary distributions, and denote by p^∞ and q^∞ their stationary distribution over $\mathcal{V}^m$, respectively.

Let $(X_{p,t})_{t\geq 1}$ (and $(X_{q,t})_{t\geq 1}$) be a Markov processes generated according to P (respectively Q). Let $Y_{p,t} = X_{p,t} - U_t$, where U_t is standard uniform random variable independent of $(X_{p,t})$ (Similarly for $Y_{q,t}$). For $t > m$, define $Z_t = F_{Y_{p,t}|X_{p,t-m:t-1}}(Y_{q,t}|X_{q,t-m:t-1})$, where $F_{Y_{p,t}|X_{p,t-m:t-1}}$ is the cumulative distribution function of Y_p conditioned on $(X_{p,t-m}, \ldots, X_{p,t-1})$. Since P is a transition kernel for m-gram Markov chain, we know $F_{Y_{p,t}|X_{p,t-m:t-1}}$ is independent of t and we can thus denote it by $F_{Y_p|m}$, which is determined by P.

Theorem 4.3.3

Let G_t be the cumulative distribution function of Z_t, then $\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} G_\tau(\cdot)$ exists and

converges to $\sum_{x_{1:m}\in\mathcal{V}^m} q^\infty(x_{1:m})F_{Y_p|k}(\cdot|x_{1:m})$ *almost surely. We also have that*

$$\sum_{x_{1:k}\in\mathcal{V}^m} q^\infty(x_{1:m})D_f(Q(\cdot|x_{1:m})\|P(\cdot|x_{1:m})) \geq D_f\left(\lim_{t\to\infty}\frac{1}{t}\sum_{\tau=1}^{t} G_\tau(\cdot)\|\mathcal{U}\right). \tag{4.9}$$

This theorem implies that the divergence measure we use in the proposed algorithm is an *asymptotically sufficient condition* for checking the difference between two Markov models P and Q.

4.4 Experiments

4.4.1 UMI in Action

Model and datasets. We now study the behavior of the UMI function on real-world large language models and datasets. In particular, we employ OLMo-1B [24] as our language model which is trained on the open Dolma dataset [66]. We use $\epsilon = 0.05$ and $\alpha = 0.1$ as the hyperparameters for the UMI function, and the Kullback-Leibler divergence as our f-divergence (See Appendix for additional evaluations of the UMI function using LLaMA2-7B [72]).

We evaluate our value function over three distinct categories of data: (1) data generated by the model using known and unknown sampling method, (2) random data consisting of tokens and (ascii) characters generated uniformly at random, (3) real-world datasets consisting of tokens from the model's training dataset and new unseen data.

Sampling Methods. In our analysis, we focused on multinomial (standard) sampling, where the next token is chosen according to the probability distribution provided by the model. However, practitioners use many other sampling methods to generate text from a language model. Thankfully, this does not change the conclusions of our analysis as all other sampling methods can be reduced to multinomial sampling on some transformed probability distribution e.g., greedy sampling, where the next token is always chosen to be the one

Dataset	Avg. Size	UMI Value(std)
Known Sampling Method	170	0.001(0.005)
Unknown Sampling Method	174	0.006(0.03)
Random Characters	815	0.026(0.073)
Random Tokens	1000	0.56(0.11)
Training Data	790	0.009(0.048)
MMLU (Unseen Data)	126	0.122(0.075)

Table 4.1: **Evaluating the UMI function on real-world data.** We report the average values returned by the UMI function, along with the standard deviation, for different categories of data, as well as the average number of tokens in the datapoints tested within each category.

with the highest probability, can be considered to be a multinomial sampling method with a distribution which places a probability of 1 on the token with the highest probability as predicted by the model.

Value of data generated by the model. The UMI function consistently assigns a very small acquisition value to the data generated by the model, regardless of the sampling method used or whether the sampling method is known. Specifically, we applied top-p sampling ($p = 0.95$) and top-k sampling ($k = 5$), but calculated the UMI function assuming standard multinomial sampling. This shows that the UMI function remains robust even when the specific sampling method and parameters are unknown.

Value of random data. We report the values returned by the UMI function when given sequences consisting of uniformly sampled ascii characters and uniformly sampled tokens. For the sequence of random ascii characters, we obtain a low value. Interestingly, the model was able to recognize that it was given a random sequence of characters and assigned near-uniform probabilities on the set of ascii characters. It is highly likely that the model was trained on such random sequences.

On the other hand, the UMI function assigned a high value to the sequence of random tokens. This is consistent with our expectations, as we anticipated a large statistical distance between the model's distribution and the i.i.d. uniform distribution over tokens. We note that this does not imply that the random tokens are useful for any particular purpose, but

rather that this sequence of tokens is very unexpected and novel from the model's perspective. In fact, an unusually high value as given by the UMI function could indicate an issue with the data as it did here.

Value of the training data. We report the values returned by the UMI function on a portion of OLMo-1B's pre-training dataset (see Appendix for additional details). We find that the UMI function assigns a higher value to the training data compared data generated by the model, but lower than data the model has not seen. This is expected, as otherwise it would indicate that the model had memorized the data it was trained on.

Value of unseen data We report the value of the value function on MMLU benchmarking dataset [32] which was not part of the model's pre-training dataset. The UMI returned a value that was higher than that of both the training dataset and the data generated by the model.

Value when the context is unknown. Language models are often used by first presenting them with a prompt and then receiving a response. It is not uncommon to run into situations where we only have access to the model's response, without knowledge of the prompt it was given. Therefore, we study the behavior of our value function when it is only provided with the model's response. In particular, given a prompt $x_{1:p}$ and response $x_{p+1:p+j}$, we evaluate our UMI function only on $x_{p+1:p+j}$ for $j \leq n$. We observe that the UMI function discerns that the tokens originate from the model despite the missing prompt (or context) and assigns it a small value (see Figure B.1 in the Appendix).

4.4.2 UMI and Training

We fine-tune LLaMA2-7B on the ViGGO dataset [38], which comprises approximately 5,000 text-to-data examples in the video game domain. Given text data in conversational form, the goal is to extract the underlying data (meaning representation). We evaluate each sample in the training set using the UMI function (with the total variation distance as our f-divergence) and rank those samples according to their computed values.

Training Set	# of Samples	Accuracy
Full Training Dataset	5103	0.762
Selected by UMI	2551	0.757
Selected by CE	2551	0.728
Random Selection	2551	0.694

Table 4.2: **Generalization Performance and UMI** We report the test accuracies obtained after fine-tuning LLaMA2-7B on training datasets selected according to different criteria.

We then compare the test accuracy of the fine-tuned model across four different training datasets: (1) the full training dataset, (2) the top 50% of the most valuable samples according to UMI, (3) the top 50% of samples with the highest cross-entropy loss, and (4) a random selection constituting 50% of the training set (repeated 5 times). For all datasets, we fine-tune for 100 epochs using AdamW with a learning rate of $\eta = 0.002$, running average coefficients $\beta = (0.99, 0.999)$, and weight decay $\lambda = 0.001$.

We find that the dataset selected according to their value as computed by UMI offers the best performance among the considered selection criteria, closely matching that of the full training dataset.

Estimating the f-divergence.

It is important to note that obtaining unbiased estimates of the f-divergence between a reference distribution and an unknown distribution using samples from the unknown distribution is a nontrivial task, highly dependent on the choice of f-divergence. However, since the reference distribution is the standard uniform distribution in this case, the problem becomes easier. In particular, when the reference distribution is $\mathcal{U}$, the Kullback-Leibler divergence simplifies to the differential entropy $h(g)$ of the unknown distribution, for which we use a nearest neighbor estimator [22, 65]. Specifically, given the samples $z_1, \ldots, z_n$ and a small fixed positive integer k, we can obtain a consistent (asymptotically unbiased) estimate using:

$$\hat{H} = -\frac{1}{n}\sum_{i=1}^{n} \ln\left(\frac{k}{2np_{k,i}}\right) - L_{k-1} + \gamma + \ln(k), \tag{4.10}$$

where $p_{i,k}$ is the distance between z_i and it's kth nearest neighbor, $L_{k-1} = \sum_{i=1}^{k-1} 1/i$ is the $(k-1)$th term of the harmonic series, and $\gamma \approx 0.5772$ is euler's constant. Moreover, when the reference distribution is $\mathcal{U}$, the total variation distance simplifies to $\int_0^1 |g(x) - 1|$, which can also be readily estimated since integration is a linear operator.

4.5 Related Works

Although this work focuses on a different problem—acquisition value—there has been significant research in data valuation over the past several years. This line of work aims to identify the training samples that most contribute to a model's performance as measured on a specific task and has largely focused on discriminative models. In [23], the 'data Shapley' is introduced for the purpose of data valuation, where the value of a datapoint is the impact on the performance of the model when that datapoint is removed from the training set. Similarly, [37] defines the consistency score (C-score) of a datapoint as the expected accuracy on a held-out datapoint given training sets sampled from a data distribution. However, to estimate the data Shapley and C-score of datapoints, the model needs to be trained multiple times. To reduce computational costs, [35] introduces more efficient algorithms when the learning algorithm satisfies certain assumptions, and suggests estimating the data Shapley through influence functions when the loss function is smooth.

[50] introduces the complexity-gap-score, a training-free quantity which acts as a proxy for the impact of individual datapoints in the optimization and generalization of classification deep neural networks. [83] avoids retraining by computing the data diversity, a data-dependent characteristic of the data itself, and connects the diversity to the learning performance. [80] estimates the performance of a network through a generalization bound that does not require training to compute, which is then used to find the value of a datapoint. [39] estimates the generalization performance of a training set by measuring the class-wise Wasserstein distance between the training and the validation sets. In [84], the authors introduce a method for data valuation specifically for generative models, where the value of a training datapoint is

determined through its similarity to a generated sample, effectively measuring the contribution of that training point to the generated output of the model.

4.6 Conclusion

In this chapter, we introduced the theoretically-grounded UMI value function, based on Rosenblatt's transformation. We analyzed this function and proved that it exactly matches the distribution distance between the model and the data in the i.i.d setting, and lower bounds the distribution distance in the Markovian setting. We evaluated our function on real-world datasets, and have showed its effectiveness in capturing the statistical distance between the model and the data. There are several open questions, including whether it is possible to integrate the semantic information of the dataset into our measure.

CHAPTER 5

Codes for 3D Orientation Estimation

5.1 Introduction

3D orientation tracking is an important function in many domains, e.g., in robotics, aerospace, and medicine. Orientation can be estimated using many methods, e.g., inertial sensors can be mounted on the object whose orientation has to be measured. Orientation can also be measured using computer vision based methods [12, 58, 69]. Each of the existing methods have their own pros and cons. For example, inertial sensors may not be suitable for many Internet-of-Things applications, e.g., tracking packages where the solutions have to be very cost-effective. Additionally, the performance of computer vision methods depends on light conditions, and can be difficult for objects that exhibit symmetries.

Wireless sensing has emerged as an interesting alternative sensing modality. Radio-frequency based methods can be useful when visible light wavelengths are not effective, e.g., cases with poor visibility or non-line-of-sight scenarios. In particular, backscatter arrays have been recently used for geo-location and 3D orientation estimation [61, 67, 68, 77]. Using backscatter arrays is philosophically akin to “painting the faces" of an object, making it a promising option for the orientation detection of symmetric objects such as a solid cube. In this work, we study 3D orientation estimation with the help of configurable backscatter arrays.

In order to aid with the estimation task, one can design the backscatter response to received signals. Specifically, we design the backscatter responses by changing their reflection

This work is supported in part by NSF grants 1955632, 2146838, 1956297, 2146838.

coefficients. Changing reflection coefficients is possible by switching between different tag load impedances [7, 71]. For example, a tag can switch between two different load impedances using a multiplexer controlled by a microcontroller (see Fig. 2 in [7]). The design of reflection coefficients can be captured as a binary code specifying what or when the backscatter tags reflect. The problem of finding the best code for the estimation task was first formulated and explored in [10], where we proposed a heuristic code design criterion and had a preliminary exploration of the systems performance.

Related work: In [10], we suggested a heuristic design criteria aimed at maximizing the average-case performance, but did not make attempts to assess its performance. Furthermore, we only considered the average-case error of the system, and did not study the worst-case error, explored in this work. In [77], the authors propose a method for estimating the 3D orientation of an object by measuring the relative phase offset between RFID tags, and aggregating the outputs of multiple 2D estimators. In [61], RFID tags are mounted on carpet pads to measure the 2D orientation of objects in an indoor environment. In [36], a 2D orientation-aware RFID approach is proposed for tracking a target. In [42], the authors track the inclination of objects using UHF RFID tags and a statistical estimator based on the received signal strength. The aforementioned works do not but do not explore the design of backscatter responses and its effect on the estimation problem. In [21], the authors use orthogonal codes (see Section 5.5 for a definition) to distinguish the tags from each other and the environment. [75, 76] use RFID tags to estimate the direction of arrival (DoA). In addition, many other works have considered the use of passive RFID tags for localization and orientation estimation [5, 59, 85, 87]. However, as far as we know, no past work has considered optimal code design for 3D orientation sensing.

In this chapter, we revisit the problem and expand our understanding in the following ways. First, we introduce two novel analytic code design criteria that take channel parameters as input, and compare their performance with respect to a baseline channel-oblivious orthogonal code. We also derive a lower bound on the worst-case error. This lower bound gives guarantees on the estimation accuracy in terms of the channel parameters including the number of

antennas and the number of tags placed on the object. We provide a comprehensive numerical exploration of the systems performance under different circumstances including when channel knowledge is not precise, multipath is present or missing. [1]

By numerically evaluating the performance, we observe the following. Our design criteria yield codes that offer significant advantages over the channel-oblivious orthogonal code for both the average and worst-case error performance (see Section 5.2.3 for precise definitions). In some cases, our designed codes offer an order of magnitude reduction in estimation error. We also find that *precise* channel knowledge at the transmitter is not critical; Our design criteria are robust against channel estimation errors, retaining a significant amounts of improvement when computed with SNR values greater than or equal to 10dB. Furthermore, our numerical results suggest that multipath offers SNR gain but not multipath-diversity gain; this means that when one fixes the total energy received across paths, the performance does not improve when multipath is present.

Organization: Section 5.2 describes the channel model and formulates the estimation problem. Section 5.3 describes the main results, including the two code design criteria. Section 5.5 provides the various numerical results. Finally, Section **??** provides proofs for the theorems and lemmas shown in Section 5.3.

5.2 Problem Formulation

5.2.1 System Model

We consider an object that can freely rotate around a specific point in space. For ease of computation, we use the 3D coordinate system in which the point the object rotates around is $\mathbf{0} = [0, 0, 0]^T \in \mathbb{R}^3$. We assume a system having K full-duplex antennas (capable of simultaneously transmitting and receiving in the same frequency band) with position vectors $\boldsymbol{x}_1^{\text{ant}}, \ldots, \boldsymbol{x}_K^{\text{ant}} \in \mathbb{R}^3$, and N backscatter tags placed on the object with position vectors

[1] Note that our method is not necessarily appropriate for real-time orientation estimation in the wild as it requires some calibration.

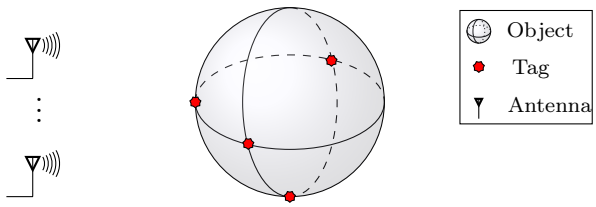

Figure 5.1: The components of the considered system. The object along with the tags, rotate around a specific point (in this case, the center of the sphere). The full-duplex antennas interrogate the configurable tags and receive back the reflected signal. We then use the received signal to infer the orientation.

$\boldsymbol{x}_1^{\text{tag}}, \ldots, \boldsymbol{x}_N^{\text{tag}} \in \mathbb{R}^3$ (see Figure 5.1).

We specify the orientation of the object using a rotation matrix $\boldsymbol{Q} \in SO(3) \subseteq \mathbb{R}^{3\times3}$, where $SO(3)$ is the 3D rotation group. If the object has tags with positions $\boldsymbol{x}_1^{\text{tag}}, \ldots, \boldsymbol{x}_N^{\text{tag}}$ on it, and a rotation $\boldsymbol{Q}$ is applied to the object, the tags would have the new positions $\boldsymbol{Q}\boldsymbol{x}_1^{\text{tag}}, \ldots, \boldsymbol{Q}\boldsymbol{x}_N^{\text{tag}}$ (see Figure 5.2). In this framework, we specify the original orientation of the object using the 3×3 identity matrix $\boldsymbol{I}_3$.

Each backscatter tag can be configured by setting its state to a value $i \in \{0, 1\}$. The state, in turn, determines the reflectivity of the tag. For instance, we may choose to correspond a state of 0 to a reflectivity of $+0.5$ and a state of 1 to a reflectivity of -0.5. In this scenario, tags reflect regardless of their state. On the other hand, we may choose to correspond a state of 0 to a reflectivity of 0 (does not reflect), and a state of 1 to a reflectivity of 1 (reflects) i.e. the state determines the on-off condition of the tag. In addition, the state of each tag can either remain constant (passive) or change over time (active). We study the performance of both options.

The orientation of the object determines the position of the tags, and the states of the tags set their reflectivities. Hence, the orientation and tag states determines the channel model, and in turn the received signal. We use the received signal to estimate the orientation of the object. Our goal is to analyze the performance of this system, and find the best set of tag states for the estimation task. We describe our channel model next.

We use the following free-space line-of-sight propagation formula [73] between points

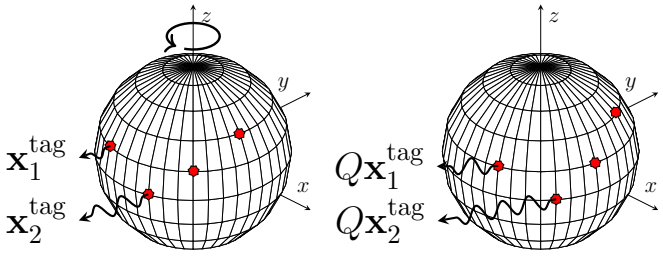

Figure 5.2: Rotation of the object about the z-axis. When the object rotates, the tags rotate along with it. The new positions of the tags produce a different received signal. While the tags are equidistant from the point of rotation in the figure, this does not have to be the case.

$\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^3$:

$$\eta(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{1}{4\pi \left\| \boldsymbol{x}_1 - \boldsymbol{x}_2 \right\|} \exp\left(\frac{-2\pi j \left\| \boldsymbol{x}_1 - \boldsymbol{x}_2 \right\|}{\lambda} \right), \tag{5.1}$$

where λ is the wavelength. In other words, suppose $\boldsymbol{x}_{\mathrm{T_x}}$ and $\boldsymbol{x}_{\mathrm{R_x}}$ are the position vectors of an isolated transmitter and receiver, and $\boldsymbol{s}_{\mathrm{T_x}}$ and $\boldsymbol{s}_{\mathrm{R_x}}$ the transmitted and received signals, respectively, then

$$\boldsymbol{s}_{\mathrm{R_x}} = \boldsymbol{s}_{\mathrm{T_x}} \eta(\boldsymbol{x}_{\mathrm{T_x}}, \boldsymbol{x}_{\mathrm{R_x}}). \tag{5.2}$$

We now specify the matrices involved in our channel model. Let $\boldsymbol{H_Q}$ be the $K \times N$ matrix of the line-of-sight responses between the tags and antennas when the object is in orientation $\boldsymbol{Q}$, i.e. $(\boldsymbol{H_Q})_{k,n} = \eta(\boldsymbol{x}_k^{\text{ant}}, \boldsymbol{Q}\boldsymbol{x}_n^{\text{tag}})$. Let $\boldsymbol{A}$ be the symmetric $K \times K$ matrix of inter-antenna line-of-sight responses with $\boldsymbol{A}_{k,k'} = \eta(\boldsymbol{x}_k^{\text{ant}}, \boldsymbol{x}_{k'}^{\text{ant}})$. Let $\boldsymbol{B}$ be the symmetric $N \times N$ matrix of the inter-tag channel responses when the object is in orientation $\boldsymbol{Q}$. In other words,

$$(\boldsymbol{B_Q})_{n,n'} = \eta(\boldsymbol{Q}\boldsymbol{x}_n^{\text{tag}}, \boldsymbol{Q}\boldsymbol{x}_{n'}^{\text{tag}}) = \eta(\boldsymbol{x}_n^{\text{tag}}, \boldsymbol{x}_{n'}^{\text{tag}}). \tag{5.3}$$

Applying the same rotation to any two tags does not change the distance between them, so we can drop $\boldsymbol{Q}$ from the subscript of $\boldsymbol{B_Q}$ and use $\boldsymbol{B}$ instead (see Figure 5.3). Now, let $\boldsymbol{s}_t \in \mathbb{C}^K$ and $\boldsymbol{s}'_t \in \mathbb{C}^N$ be the vectors of the transmitted signals at the antennas and tags respectively at time t, and $\boldsymbol{\Psi}_t \in \mathbb{C}^K$ and $\boldsymbol{\Psi}'_t \in \mathbb{C}^N$ be the vectors of the received signals at the antennas and tags respectively at time t. The relationship between these vectors is given

by:

$$\begin{bmatrix} \boldsymbol{\Psi}_t \\ \boldsymbol{\Psi}'_t \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{H_Q} \\ \boldsymbol{H}_{\boldsymbol{Q}}^T & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_t \\ \boldsymbol{s}'_t \end{bmatrix}. \tag{5.4}$$

As we have mentioned before, each backscatter tag can be configured by setting its state to a value $i \in \{0, 1\}$, which in turn determines its reflectivity $r_i \in \mathbb{C}$. Hence, the reflectivities can be expressed in terms of the states assigned to the tags. Letting $s_{t,n} \in \{0, 1\}$ be the state of the n^{th} tag at time t, $\boldsymbol{c}_t = (s_{t,1}, \ldots, s_{t,N})$ be the *codeword* at time t (a codeword is a vector of states), and $\boldsymbol{r}(\cdot)$ be the mapping from codewords to reflectivity vectors, we define the matrix of reflectivities at time t as:

$$\boldsymbol{R}_t = \text{diag}(\boldsymbol{r}(\boldsymbol{c}_t)) \in \mathbb{R}^{N \times N}. \tag{5.5}$$

Using the fact that the tags reflect the incident signals, the transmitted signal at the tags is given by $\boldsymbol{s}'_t = \boldsymbol{R}_t \boldsymbol{\Psi}'_t$. We note that the tags also contribute an additive term. A typical RFID reader can easily filter out this component by the receiver's ac-coupling capacitor or DC-offset compensation loop [7] [71]. Thus, we ignore this constant term in our model. Substituting the value of $\boldsymbol{s}'_t$ in (5.4), and adding the complex Gaussian noise terms $\boldsymbol{w}_t \in \mathbb{C}^K$ and $\boldsymbol{w}'_t \in \mathbb{C}^N$, we obtain the following equations:

$$\boldsymbol{\Psi}_t = \boldsymbol{A}\boldsymbol{s}_t + \boldsymbol{H_Q}\boldsymbol{R}_t\boldsymbol{\Psi}'_t + \boldsymbol{w}_t \tag{5.6}$$

$$\boldsymbol{\Psi}'_t = \boldsymbol{H}_{\boldsymbol{Q}}^T\boldsymbol{s}_t + \boldsymbol{B}\boldsymbol{R}_t\boldsymbol{\Psi}'_t + \boldsymbol{w}'_t \tag{5.7}$$

Solving the above equations, we obtain that the received signal is given by $\boldsymbol{\Psi}_t = \boldsymbol{A}\boldsymbol{s}_t + \boldsymbol{H_Q}\boldsymbol{R}_t(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{R}_t)^{-1}\left(\boldsymbol{H}_{\boldsymbol{Q}}^T\boldsymbol{s}_t + \boldsymbol{w}'_t\right) + \boldsymbol{w}_t$. We assume that the noise term involving $\boldsymbol{w}'_t$ is dominated by $\boldsymbol{w}_t$, and hence we ignore the former. Since the first term in the previous expression does not depend on $\boldsymbol{Q}$, we can subtract it at the receiver[2] yielding a modified

[2]We are eliminating full-duplex self-interference, *e.g.,* see [57].

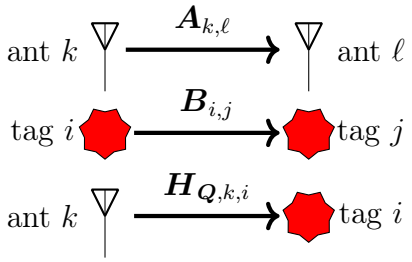

Figure 5.3: The matrices associated with the model. For instance, $A_{k,\ell}$ corresponds to the free space loss *from* antennas k *to* antenna ℓ.

noiseless and noisy signals $\boldsymbol{f}(\boldsymbol{Q};\boldsymbol{c}_t), \boldsymbol{y}(\boldsymbol{Q};\boldsymbol{c}_t) \in \mathbb{C}^K$ given by:

$$\boldsymbol{f}(\boldsymbol{Q};\boldsymbol{c}_t) = \boldsymbol{H}_{\boldsymbol{Q}}\boldsymbol{R}_t(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{R}_t)^{-1}\boldsymbol{H}_{\boldsymbol{Q}}^T\boldsymbol{s}_t, \tag{5.8}$$

$$\boldsymbol{y}(\boldsymbol{Q};\boldsymbol{c}_t) = \boldsymbol{f}(\boldsymbol{Q};\boldsymbol{c}_t) + \boldsymbol{w}_t. \tag{5.9}$$

5.2.2 Codes

The expression in (5.8) is the *noiseless* channel output when using a codeword $\boldsymbol{c}_t$. Based on numerical simulations, we have observed that a single codeword often only allows us to detect some orientations, and not others. Specifically, all tag arrays we came across had the following property: for every codeword $\boldsymbol{c} \in \{0,1\}^N$, there exists some $\boldsymbol{Q}, \boldsymbol{Q}'$ such that $\boldsymbol{Q} \neq \boldsymbol{Q}'$ and $\boldsymbol{f}(\boldsymbol{Q};\boldsymbol{c}) \approx \boldsymbol{f}(\boldsymbol{Q}';\boldsymbol{c})$ (see Figure 5.5 in Section 5.5 for such an array). In other words, for every codeword, there is a pair of orientations that the codeword makes almost indistinguishable based on the received signal.

Therefore, using different codewords over time is a natural extension to the model as a new codeword could fill in the gaps left by the ones used previously. Based on this intuition, we define a *code* $\boldsymbol{C} = [\boldsymbol{c}_1 \cdots \boldsymbol{c}_T] \in \{0,1\}^{N\times T}$ to be a binary matrix where each column corresponds to the codeword at a specific time t (See Figure 5.4). Given the code $\boldsymbol{C}$, we denote the *concatenated* noiseless and noisy channel outputs as (using $\boldsymbol{F}$ instead of $\boldsymbol{f}$ and $\boldsymbol{Y}$

instead of $\boldsymbol{y}$):

$$\boldsymbol{F}(\boldsymbol{Q};\boldsymbol{C}) = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{Q};\boldsymbol{c}_1) & \cdots & \boldsymbol{f}(\boldsymbol{Q};\boldsymbol{c}_T) \end{bmatrix} \in \mathbb{C}^{K\times T}, \tag{5.10}$$

$$\boldsymbol{Y}(\boldsymbol{Q};\boldsymbol{C}) = \boldsymbol{F}(\boldsymbol{Q};\boldsymbol{C}) + \boldsymbol{W} \in \mathbb{C}^{K\times T}, \tag{5.11}$$

where $\boldsymbol{W} = [\boldsymbol{w}_1 \cdots \boldsymbol{w}_t] \in \mathbb{C}^{K\times T}$ is complex Gaussian noise.

5.2.3 Performance Measures

We consider estimation over a finite uniform subset of orientations $\mathcal{Q} \subset \mathrm{SO}(3)$, and use the loss function $\theta(\boldsymbol{Q}, \boldsymbol{Q}')$, equal to the Frobenius norm between the two rotation matrices i.e.,

$$\theta(\boldsymbol{Q}, \boldsymbol{Q}') = \left\| \boldsymbol{Q} - \boldsymbol{Q}' \right\|_F . \tag{5.12}$$

Letting $\boldsymbol{C} \in \{0,1\}^{N\times T}$ be any code, the expected value of the error of $\boldsymbol{C}$ when the ground-truth orientation of the object is $\boldsymbol{Q} \in \mathcal{Q}$ is given by:

$$\mathcal{L}(\boldsymbol{Q}, \boldsymbol{C}) = \mathbb{E}_{\boldsymbol{W}} \left[\theta \left(\widehat{\boldsymbol{Q}} \left(\boldsymbol{Y}(\boldsymbol{Q};\boldsymbol{C}) \right), \boldsymbol{Q} \right) \right]. \tag{5.13}$$

where $\widehat{\boldsymbol{Q}} \left(\boldsymbol{Y}(\boldsymbol{Q};\boldsymbol{C}) \right)$ is the minimum distance decoder to the grid of orientations $\mathcal{Q}$ i.e., the MMSE estimator of $\boldsymbol{Q}$ given $\boldsymbol{Y}(\boldsymbol{Q};\boldsymbol{C})$. Assuming a uniform prior over the orientations in $\mathcal{Q}$; i.e., the object is equally likely to be in any of the orientations in $\mathcal{Q}$, the expected value of the error of code $\boldsymbol{C}$ (w.r.t. to the orientations) is given by:

$$\mathcal{L}(\boldsymbol{C}) = \frac{1}{|\mathcal{Q}|} \sum_{\boldsymbol{Q}\in\mathcal{Q}} \mathcal{L}(\boldsymbol{Q}, \boldsymbol{C}), \tag{5.14}$$

Now, a code that minimizes the average error (w.r.t. to the orientations) could still provide poor performance for a *particular* orientation. In other words, the performance of a code $\boldsymbol{C}$ might vary from one orientation to the next. For this reason, we study the worst-case error as it delivers guarantees on the performance across every orientation. We define worst-case

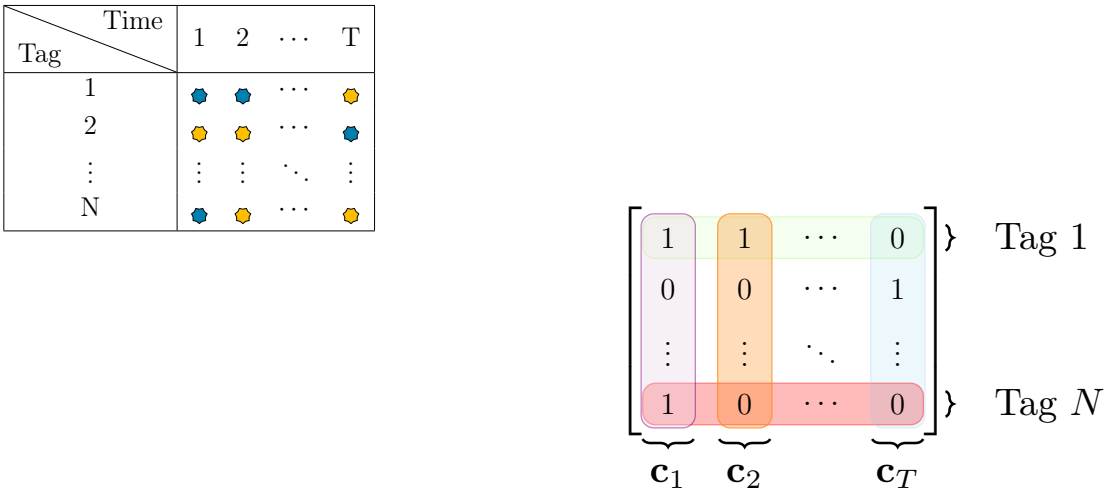

Figure 5.4: An illustration of coding over time (left) and its corresponding code (right). Different colors denote different states (reflectivities) of the tag, and different patterns of the states denote different codewords.

error (over all possible orientations in $\mathcal{Q}$) as:

$$\mathcal{M}(\boldsymbol{C}) = \max_{\boldsymbol{Q} \in \mathcal{Q}} \mathcal{L}(\boldsymbol{Q}, \boldsymbol{C}). \tag{5.15}$$

We believe that both quantities are of interest for designing an orientation estimation system and hence, we suggest criteria for minimizing either depending on the application's particular requirements.

5.3 Theoretical Analysis

Our goal is to find the best code for the estimation task. However, $\mathcal{L}(\boldsymbol{C})$ in (5.14) and $\mathcal{M}(\boldsymbol{C})$ in (5.15) are intractable. One route we could take is to investigate *tractable* upper or lower bounds on the errors, and minimize those bounds as a proxy for minimizing the errors themselves.

5.4 Design Criterion For The Average Error

As a proxy for the average error $\mathcal{L}(\boldsymbol{C})$, we minimize an upper bound to it. We obtain the upper bound on the average error by replacing the probability of misestimating the actual orientation of the object by a tractable quantity. The proof of the following result is given in Section C.1.1.

Theorem 5.4.1

Assuming complex white Gaussian noise, the average error $\mathcal{L}(\boldsymbol{C})$ *given in* (5.14) *is upper bounded by:*

$$\mathcal{U}(\boldsymbol{C}) = \sum_{\boldsymbol{Q},\boldsymbol{Q}' \in \mathcal{Q}} \operatorname{erfc}\left(\frac{\|\boldsymbol{F}(\boldsymbol{Q};\boldsymbol{C}) - \boldsymbol{F}(\boldsymbol{Q}';\boldsymbol{C})\|_F}{2\sqrt{2}\sigma}\right) \theta(\boldsymbol{Q},\boldsymbol{Q}'). \tag{5.16}$$

where $\theta(\boldsymbol{Q},\boldsymbol{Q}')$ *is given in* (5.12), erfc *is the complement of the Gaussian error function, and* σ *is the standard deviation of the noise.*

Hence, the design criteria for minimizing the average error is simply given by the optimization problem $\min_{\boldsymbol{C}} \mathcal{U}(\boldsymbol{C})$. We can infer the following intuition from $\mathcal{U}(\boldsymbol{C})$ in (5.16): good codes should map orientations that are far apart (in terms of θ) to channel outputs that are also far apart. Moreover, since $\operatorname{erfc}(x)$ is tightly upper bounded by $\exp(-x^2)$, $\mathcal{U}(\boldsymbol{C})$ implies that the distance between channel outputs $\|\boldsymbol{F}(\boldsymbol{Q};\boldsymbol{C}) - \boldsymbol{F}(\boldsymbol{Q}';\boldsymbol{C})\|_F$ has an inverse exponential relationship with the average error.

5.4.1 Codes For the Worst-Case Error

Whereas we optimize an upper bound for the average error in Section 5.4, we obtain a lower bound on the worst-case error $\mathcal{M}(\boldsymbol{C})$ using Le Cam's method [62], and minimize the bound to obtain a code for the worst-case performance. The proof of the following theorem is provided in Section C.1.2.

Theorem 5.4.2

Assuming complex white Gaussian noise, the worst-case error $\mathcal{M}(\boldsymbol{C})$ given in (5.15) is lower bounded by:

$$\mathcal{V}(\boldsymbol{C}) = \max_{\boldsymbol{Q},\boldsymbol{Q}'\in\mathcal{Q}} \exp\left(-\frac{\|\boldsymbol{F}(\boldsymbol{Q};\boldsymbol{C}) - \boldsymbol{F}(\boldsymbol{Q}';\boldsymbol{C})\|_F^2}{2\sigma^2}\right)\frac{\theta(\boldsymbol{Q},\boldsymbol{Q}')}{4}, \tag{5.17}$$

where $\theta(\boldsymbol{Q},\boldsymbol{Q}')$ is given in (5.12).

Hence, the design criteria for minimizing the worst-case error is given by the optimization problem $\min_C \mathcal{V}(\boldsymbol{C})$. $\mathcal{V}(\boldsymbol{C})$ corroborates the observations from Theorem 5.4.1, and relates the worst-case error to the model only through the distances between channel outputs. $\mathcal{V}(\boldsymbol{C})$ again implies that the worst-case error has an inverse exponential relationship with $\|\boldsymbol{F}(\boldsymbol{Q};\boldsymbol{C}) - \boldsymbol{F}(\boldsymbol{Q}';\boldsymbol{C})\|_F$.

5.4.2 Minimax Bound

Unlike Theorem 5.4.2 which suggests a design criteria for the worst-case error using Le Cam's method, the following theorem uses the same tools to quantify the worst-error error decay with parameters including the number of antennas K, the number of tags N through the Frobenius norm of $\boldsymbol{X}^{\text{tag}}$, and the number of samples through the variance σ^2 (as the effective variance of the noise is σ^2/n when provided with n samples). It also reflects the effect of code design on the error.

Theorem 5.4.3 (Minimax Bound)
The worst-case error is bounded as:

$$\mathcal{M}(\boldsymbol{C}) \geq \frac{32\pi^2\lambda^2\sigma^2 D^4}{27K^2\left\|\boldsymbol{X}^{\text{tag}}\right\|_F^2 \sum_{t=1}^{T}\|\widetilde{\boldsymbol{B}}_t\|_F^2\left\|\boldsymbol{r}(\boldsymbol{c}_t)\right\|^2} \tag{5.18}$$

where $\widetilde{\boldsymbol{B}}_t = (\boldsymbol{I} - \boldsymbol{B}\boldsymbol{R}_t)^{-1}$ and D is an approximate range between the tagged object and the antennas.

The bound indicates that placing the tags far away from the antennas leads to a larger

error. This is clear as the farther away the tags, the weaker the signal we receive due to attenuation. Moreover, it implies an inverse quadratic relationship between the number of antennas and the error. This is again logical because we would be able to receive more power from the reflected signal with more antennas. The bound also reflects the effect of code design through the term $\|\tilde{\boldsymbol{B}}_t\|_F^2 \|\boldsymbol{r}(\boldsymbol{c}_t)\|^2$. For instance, if tags can each have one of two states with corresponding reflectivities 0 and 1 (on or off), then the bound favors most tags to be "on" through the term $\|\boldsymbol{r}(\boldsymbol{c}_t)\|^2$.

5.4.3 Structure of the optimal codes

Finding the codes that minimize $\mathcal{U}(\boldsymbol{C})$ and $\mathcal{V}(\boldsymbol{C})$ is a combinatorial optimization problem that requires an exhaustive search. Performing an exhaustive search has a running time of $\mathcal{O}(2^{NT})$ if wish to look for the best coding matrix of size T. Hence, for our purposes, finding the best code using brute-force is intractable except for cases when the number of tags N, and the size of the coding matrix T, are very small.

Therefore, we introduce the idea of code proportions which is essential in reducing the search space. We show that our performance measures depend only on the code through particular proportions of each codeword (see Theorem 5.4.4), therefore reducing the search space to only those proportions. We begin by carefully defining this idea. Let $\boldsymbol{C} = [\boldsymbol{c}_1 \cdots \boldsymbol{c}_T]$ be any code, then we define the proportion of code configuration $\boldsymbol{c} \in \{0,1\}^N$ in $\boldsymbol{C}$ as

$$\boldsymbol{\pi}_{\boldsymbol{c}} = \frac{1}{T} \sum_{t=1}^{T} 1_{\boldsymbol{c}=\boldsymbol{c}_t}, \tag{5.19}$$

where $1_{\boldsymbol{c}=\boldsymbol{c}_t}$ is the indicator of the event $\{\boldsymbol{c} = \boldsymbol{c}_t\}$. In other words, $\boldsymbol{\pi}_{\boldsymbol{c}}$ is the number of occurrences of $\boldsymbol{c}$, normalized by T, the size of the code. Moreover, let $\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(2^N)}$ be an enumeration of the codewords in $\{0,1\}^N$, then we define the vector of code proportions as $\boldsymbol{\pi} = \left[\pi_{\boldsymbol{c}^{(1)}}, \cdots, \pi_{\boldsymbol{c}^{(2^N)}}\right]^T$. It is straightforward to see that if two coding matrices share the same size T, and the same proportions vector $\boldsymbol{\pi}$, then they are permutations of one another. In what follows, instead of directly using the coding matrix $\boldsymbol{C}$, we search for the optimal

coding scheme through the proportions vector $\boldsymbol{\pi}$.

The following theorem allows us to analyze the effect of $\boldsymbol{C}$ on the average and worst-case errors through the proportions vector $\boldsymbol{\pi}$. Moreover, we show that the derived bounds in (5.16) and (5.17) can be written in terms of $\boldsymbol{\pi}$. The proofs of the following results is given in Section C.1.6.

Theorem 5.4.4

Assuming independent and identically distributed noise, $\mathcal{L}(\boldsymbol{C})$ and $\mathcal{M}(\boldsymbol{C})$ only depend on coding matrix $\boldsymbol{C}$ through the proportions vector $\boldsymbol{\pi}$.

Lemma 5.4.1

If we fix T, the size of the code $\boldsymbol{C}$, then we can write the minimization of $\mathcal{U}(\boldsymbol{C})$ and $\mathcal{V}(\boldsymbol{C})$ in term of $\boldsymbol{\pi}$ respectively as:

$$\mathcal{U}(\boldsymbol{\pi}) = \sum_{\boldsymbol{Q},\boldsymbol{Q}' \in \mathcal{Q}} \operatorname{erfc}\left(\sqrt{\frac{T\left\langle \boldsymbol{\pi}, \boldsymbol{g}(\boldsymbol{Q},\boldsymbol{Q}')\right\rangle}{8\sigma^2}}\right)\theta(\boldsymbol{Q},\boldsymbol{Q}'), \tag{5.20}$$

$$\mathcal{V}(\boldsymbol{\pi}) = \max_{\boldsymbol{Q},\boldsymbol{Q}' \in \mathcal{Q}} \exp\left(-\frac{T\left\langle \boldsymbol{\pi}, \boldsymbol{g}(\boldsymbol{Q},\boldsymbol{Q}')\right\rangle}{2\sigma^2}\right)\theta(\boldsymbol{Q},\boldsymbol{Q}'), \tag{5.21}$$

where $\boldsymbol{g}(\boldsymbol{Q},\boldsymbol{Q}') \in \mathbb{R}^{2^N}$ is given by $\boldsymbol{g}(\boldsymbol{Q},\boldsymbol{Q}')_i = \left\|\boldsymbol{f}(\boldsymbol{Q};\boldsymbol{c}^{(i)}) - \boldsymbol{f}(\boldsymbol{Q}';\boldsymbol{c}^{(i)})\right\|$.

Theorem 5.4.4 and Lemma 5.4.1 allow us to write our design criteria as optimization problems with respect to the proportions vector $\boldsymbol{\pi}$. In (5.20) and (5.21), $\boldsymbol{g}(\boldsymbol{Q},\boldsymbol{Q}')$ represents the link between the design criteria and the channel model. This implies that our design criteria will depend on the channel model only through the distances between channel outputs for different values of the orientation. In other words, the design criteria for the average-case and worst-case errors are respectively given by:

$$\begin{aligned} \min_{\boldsymbol{\pi}\in\mathbb{R}^{2^N}} \quad & \sum_{\boldsymbol{Q},\boldsymbol{Q}' \in \mathcal{Q}} \operatorname{erfc}\left(\sqrt{\frac{T\left\langle \boldsymbol{\pi}, \boldsymbol{g}(\boldsymbol{Q},\boldsymbol{Q}')\right\rangle}{8\sigma^2}}\right)\theta(\boldsymbol{Q},\boldsymbol{Q}') \\ \text{s.t.} \quad & \sum_{i=1}^{2^N} \boldsymbol{\pi}_i = 1, \boldsymbol{\pi} \geq 0. \end{aligned} \tag{5.22}$$

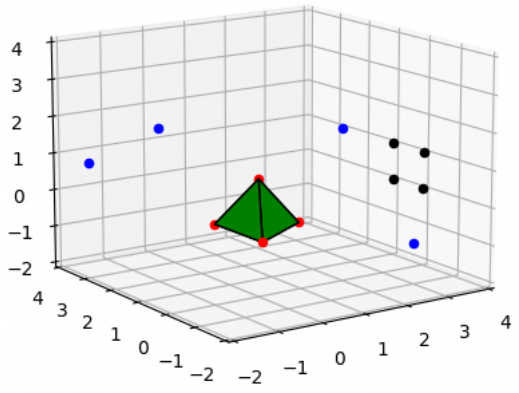

Figure 5.5: An example of a simulation setup. The tags (given by the red dots) are in a tetrahedron configuration, while the antennas (given by the black dots) are in a square configuration. Reflectors are denoted by the blue dots.

$$\begin{aligned} \min_{\boldsymbol{\pi}\in\mathbb{R}^{2^N}} \quad & \max_{\boldsymbol{Q},\boldsymbol{Q}'\in\mathcal{Q}} \exp\left(-\frac{T\left\langle \boldsymbol{\pi}, \boldsymbol{g}(\boldsymbol{Q},\boldsymbol{Q}')\right\rangle}{2\sigma^2}\right)\theta(\boldsymbol{Q},\boldsymbol{Q}') \\ \text{s.t.} \quad & \sum_{i=1}^{2^N} \boldsymbol{\pi}_i = 1, \boldsymbol{\pi} \geq 0. \end{aligned} \tag{5.23}$$

5.5 Numerical Simulations

5.5.1 Simulation Setup

We use $N = 4$ backscatter tags that we place randomly within a sphere around the center. The 4 tags can *each* have two states, with reflection coefficients -0.5 and 0.5, encoded as 0 and 1, respectively. We arrange $K = 4$ full-duplex antennas in a 1m $\times$ 1m square on a plane 4m away from the center of the object (see Figure 5.5 for a sample arrangement). Each antenna emits an identical signal $\boldsymbol{s}_k = 1$ for all $k = 1, \ldots, K$. We use a wavelength $\lambda = 0.005$m, and generate a set of orientations $\mathcal{Q}$ with $|\mathcal{Q}| = 4096$ by uniformly sampling the ranges of the euler angles and computing the rotation matrices that correspond to the angles.

To find the solution of (5.22) and (5.23), we use the well-known trust-region method (TRM) [86]. The trust-region method first defines a region around the current solution, and approximates the original objective function using a quadratic model. TRM then takes a

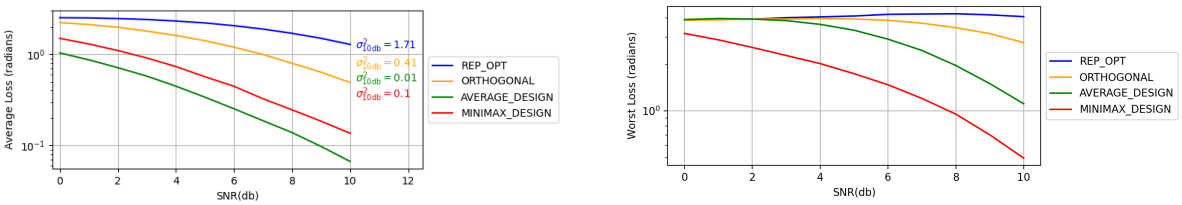

Figure 5.6: The average (top) and worst-case (bottom) errors versus SNR for the tag array in Figure 5.5. At an SNR value of 10dB, the designed code for the average error is nearly 8× more accurate than the channel-agnostic orthogonal code. Meanwhile, the designed code for the worst-case error is nearly 10× more accurate.

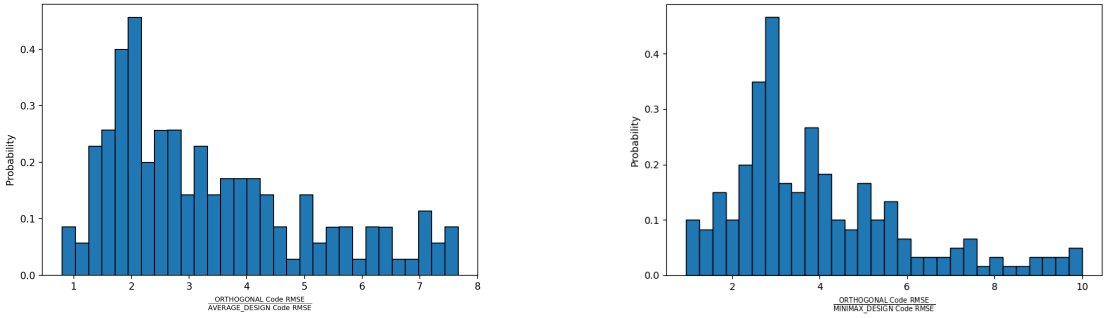

Figure 5.7: Probability density histograms for the average error (top) and worst-case error (bottom) ratios when using the orthogonal code and the designed code at an SNR of 10dB.

step according to what quadratic model depicts in the region. Computing the quadratic approximation to the original objective function grows more computationally demanding as we increase the size $\mathcal{Q}$. However, instead of decreasing the size of $\mathcal{Q}$, we chose to keep $\mathcal{Q}$ as is and use a uniform subset of the orientations in $\mathcal{Q}$ to solve (5.22), (5.23). In particular, for our simulations, we used a set of orientations $\mathcal{Q}' \subset \mathcal{Q}$ of size 200. The optimization problem requires 5 to 10 minutes to converge on a personal machine. We define SNR as the *received signal-to-noise ratio*, the received signal strength at each antenna, divided by the noise power. In other words $\text{SNR} = 10\log_{10}\left(\frac{\|\boldsymbol{F}\|_2^2}{\sigma^2}\right)$, where $\boldsymbol{F}$ is the noiseless signal and σ^2 is the noise variance. We test each value of $\text{SNR} = 0, 1, \ldots, 12$ dB, with trials for each of the 4096 "ground truth" orientations in $\mathcal{Q}$. As a measure of distance, we use the polar and azimuthal angles in the polar coordinate system (r, θ, φ). All things considered, the measure of distance we use in our trials is given by:

$$\ell\left[(\theta,\varphi),(\hat{\theta},\hat{\varphi})\right] = (\theta-\hat{\theta})^2 + (\varphi-\hat{\varphi})^2, \tag{5.24}$$

where (θ,φ) are the ground truth angles, and $(\hat{\theta},\hat{\varphi})$ are the estimated angles. To obtain an estimate $\widehat{\mathcal{L}}(\boldsymbol{C})$ of the average-case error $\mathcal{L}(\boldsymbol{C})$, we average the value of the error over the trials, and compute root-mean-square error (RMSE). The performance of a code varies across different orientations. Hence, unique to average error trials, we have included the variance of each coding method at an SNR value of 10dB. The variance indicates how much the performance varies from the total average across the different tested orientations. On the other hand, to obtain an estimate $\widehat{\mathcal{M}}(\boldsymbol{C})$ of the worst-case error $\mathcal{M}(\boldsymbol{C})$, we average the 500 trials for each of the considered orientations, and then take the maximum over all the orientations in $\mathcal{Q}$. In these simulation trials, we use codes of size $T = 24$, and compare the performance of the following methods:

- REP_OPT: The best repetition code (all $T = 24$ codewords used are the same). This method is equivalent to the design criteria suggested in [10].
- ORTHOGONAL: The code where each of the vectors in the standard basis $\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_4\}$ is used 6 times. Note that $\boldsymbol{e}_i$ is the vector components whose are all zero, except for a 1 in the i^{th} component.
- AVERAGE_DESIGN: The method we suggested to minimize the average error. This is the coding scheme that minimizes the upper bound of the average error i.e. $\boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}} \mathcal{U}(\boldsymbol{\pi})$.
- MINIMAX_DESIGN: The method we suggested to minimize the worst-case error. This is the coding scheme that minimizes the lower bound of the worst-case error i.e. $\boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}} \mathcal{V}(\boldsymbol{\pi})$.

5.5.2 The Average and Worst-Case Errors

The results shown in Figure 5.6 suggest that channel knowledge at the receiver provides significant benefits over orthogonal codes. For the average error, the code we designed, AVERAGE_DESIGN, lead to the best performance out of the considered methods, with the code produced by the method we suggested for the worst-case error, MINIMAX_DESIGN, trailing behind as a close second. In particular, for the tetrahedron tag configuration shown in Figure 5.5, the average error when using the code produced by the optimization problem $\min_{\boldsymbol{\pi}} \mathcal{U}(\boldsymbol{\pi})$ is approximately $8\times$ lower than the error when using orthogonal code. Furthermore, the performance obtained when using the best static code is consistently the worst out of the considered methods. This corroborates the intuition that lead us to introduce the idea of different codes over time. Moreover, the results shown in Figure 5.6 (bottom) suggest that channel knowledge for code design once again provides benefits for the worst-case error. For the tetrahedron array of Figure 5.5, the worst-case error produced by the optimization problem $\min_{\boldsymbol{\pi}} \mathcal{V}(\boldsymbol{\pi})$ is approximately $10\times$ lower than the error produced when using orthogonal codes.

In order to test the effectiveness of the designed criteria on tag arrays different from the one shown in Figure 5.5, we randomize the positions of the tags (within a sphere of radius 0.25m), compute the designed codes according to the methods of AVERAGE_DESIGN and MINIMAX_DESIGN, and calculate the average and worst-case error ratios obtained when using orthogonal codes versus the designed codes. We perform these steps for a total of 100 times, and produce the density histogram shown in Figure 5.7. In many of the 100 trials, the designed codes offered more a significant improvement over the channel oblivious orthogonal code for the average error (Figure 5.7 top) and worst-case error (Figure 5.7 bottom).

5.5.3 Multipath

Suppose we now add M reflectors with positions $\boldsymbol{x}_1^{\text{ref}}, \ldots, \boldsymbol{x}_M^{\text{ref}} \in \mathbb{R}^3$ to our system. We treat reflectors as virtual sources. For an antenna with position $\boldsymbol{x}_k^{ant}$, tag with position $\boldsymbol{x}_n^{tag}$, and

reflector with position $\boldsymbol{x}_m^{\text{ref}}$, the ray can take several possible paths: (1) the path that starts at the antenna, passes through the reflector, and ends at the tag, and back the same path, (2) from the antenna to the tag through the LoS path, and then back along the path of the reflector and vice versa. To capture these paths in our model, we define

$$\gamma(\boldsymbol{x}_k^{ant}, \boldsymbol{x}_m^{\text{ref}}, \boldsymbol{x}_n^{tag}) = \eta(\boldsymbol{x}_k^{ant}, \boldsymbol{x}_m^{\text{ref}})\eta(\boldsymbol{x}_m^{\text{ref}}, \boldsymbol{x}_n^{tag}). \tag{5.25}$$

Let $\boldsymbol{D}_{\boldsymbol{Q},m}$ be the $K \times N$ matrix of the multipath only channel responses between the tags and antennas in the presence of $\boldsymbol{x}_m^{\text{ref}}$, i.e.

$$(\boldsymbol{D}_{\boldsymbol{Q},m})_{k,n} = \gamma(\boldsymbol{x}_k^{\text{ant}}, \boldsymbol{x}_m^{\text{ref}}, \boldsymbol{Q}\boldsymbol{x}_n^{\text{tag}}). \tag{5.26}$$

The matrices involved in our channel model remain largely the same, except for $\boldsymbol{H}_{\boldsymbol{Q}}$ which is now replaced by $\boldsymbol{E}_{\boldsymbol{Q}} = \boldsymbol{H}_{\boldsymbol{Q}} + \boldsymbol{D}_{\boldsymbol{Q}}$, where $\boldsymbol{D}_{\boldsymbol{Q}} = \sum_{m=1}^{M} \boldsymbol{D}_{\boldsymbol{Q},m}$ The noiseless channel model is thus extended to:

$$\boldsymbol{f}_m(\boldsymbol{Q}; \boldsymbol{c}_t) = \boldsymbol{E}_{\boldsymbol{Q}}\boldsymbol{R}_t(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{R}_t)^{-1}\boldsymbol{E}_{\boldsymbol{Q}}^T\boldsymbol{s}_t. \tag{5.27}$$

In our trials, we use 10 multipath components placed 1m away from the object. Based on observations, and the results of Figure 5.8, multipath does add a small diversity gain. Moreover, given a fixed transmit power, multipath does also offer an SNR gain at the receiver.

5.5.4 Robustness

Computing the codes schemes suggested by our design criteria requires channel knowledge. However, we are not likely to have perfect channel knowledge, and so we test the performance of our design criteria when computed using imperfect channel estimation. We introduce channel estimation errors through several methods: (1) adding isotropic noise to channel outputs, (2) adding noise to the inter-tag channel response matrix $\boldsymbol{B}$, (3) misestimating the reflections coefficients of the tags.

The simplest of introducing channel estimation errors is adding isotropic noise to the

channel outputs in different orientations. Specifically, instead of using $\boldsymbol{F}(\boldsymbol{Q};\boldsymbol{C})$ to compute the suggested coding scheme, we use

$$\boldsymbol{F}'(\boldsymbol{Q};\boldsymbol{C}) = \boldsymbol{F}(\boldsymbol{Q};\boldsymbol{C}) + \boldsymbol{W}_{\boldsymbol{Q}}, \tag{5.28}$$

where $\boldsymbol{W}_{\boldsymbol{Q}}$ is white Gaussian noise, and $\boldsymbol{W}_{\boldsymbol{Q}}$ is independent of $\boldsymbol{W}_{\boldsymbol{Q}'}$ for every $\boldsymbol{Q}' \neq \boldsymbol{Q}$. We tested the following values of the SNR $= 5, 10, 15$dB, with 50 trials each, and then averaged the performance. In particular, the results of Figure 5.9 indicate that our designed codes retain nearly all even when computed with channel estimation errors greater than or equal to 10dB. The figures also indicate that the design criteria maintain considerable improvement over channel oblivious codes when computed with channel estimation errors less than 10dB.

On the other hand, we can add channel estimation errors by adding noise to the the inter-tag channel response matrix $\boldsymbol{B}$ (see Equation (5.3)). The noise used is normalized by the entries of $\boldsymbol{B}$. The results of Figure C.1 indicate that misestimating $\boldsymbol{B}$ has a minor effect on the performance when $\boldsymbol{B}$ is misestimated with errors less than or equal to 10dB. Moreover, we add channel errors by misestimating the reflectivity of the tags (e.g. estimating the reflectivity as $+0.4, -0.4$ when the actual reflectivities are $+0.5, -0.5$). We measure the misestimation of tag reflection coefficients using the relative absolute error i.e., if we assume true reflectivities $0.5, -0.5$ and a relative error of 10%, then we compute the codes under the assumption that the reflectivities are $\{(0.4, -0.4), (0.6, -0.6), (0.4, -0.6), (0.6, -0.4)\}$.

5.6 Conclusion

Active backscatter tags can be very useful when estimating the 3D orientation of objects if tag responses are carefully designed. In this work, we suggested two design criteria, and developed a bound showing how different system parameters affect performance. Through numerical simulations, we exhibited the effectiveness of the suggested designs, and their robustness in the face of imperfect channel knowledge. There are several open questions remaining including designs for more complicated or time-varying environments. Additionally,

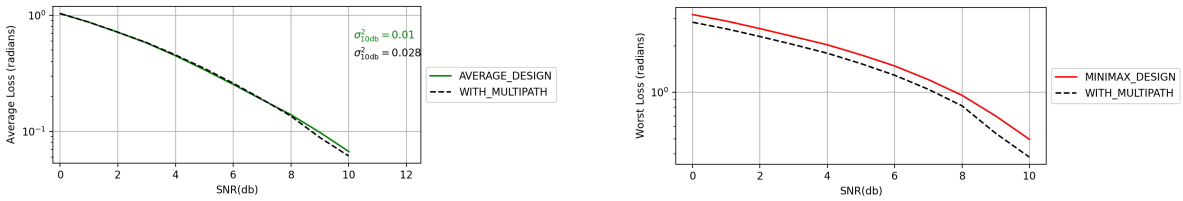

Figure 5.8: The average (top) and worst-case (bottom) errors for the designed codes in the cases of line-of-site only and multipath.

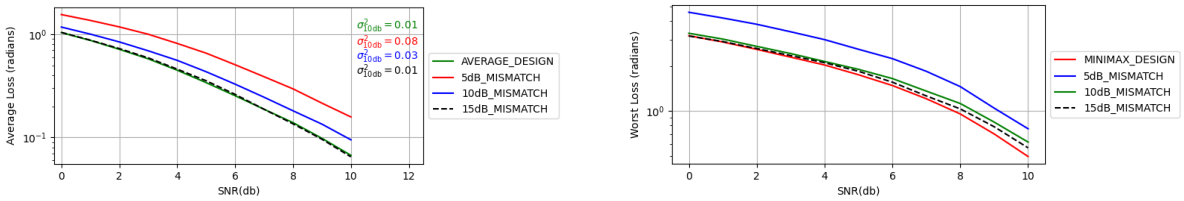

Figure 5.9: The average (top) and worst-case (bottom) errors of the designed codes for different levels of channel estimation errors.

an exploration of the effectiveness of non-coherent coding in such scenarios is a potentially interesting direction.

CHAPTER 6

Conclusion and Future Work

In this dissertation, we have studies the problems of deep learning generalization, data valuation for generative models, and 3D orientation estimation from an information-theoretic perspective. Through the development of novel algorithms inspired from theory and rigorous mathematical analysis, we have demonstrated that the usefulness of information theory is not constrained to the communication only, but extends farther to include various disparate topics and fields. In this chapter, we reiterate the contribution made in this dissertation and discuss directions for future work.

6.1 Information-Theoretic Generalization Bounds

In chapter 3, we introduced a bound based on the new information-theoretic measure of leave-one-out conditional mutual information. Bounds based on the leave-one-out conditional mutual information reduce computational costs, connect to other quantities present in the generalization theory literature such as stability, and improve under the interpolation assumption. We found that it provides non-vacuous guarantees in the image classification tasks we evaluated it on. There are several open questions and future directions. For instance, we ask whether it is possible to maintain the tighter bound that interpolating algorithms enjoy under weaker assumption on the learning algorithm, or even for general machine learning algorithms. Moreover, we ask whether it is possible to remove the need for retraining the model when computing the bound. For some machine learning algorithms such as linear regression, one can obtain the channel output when one datapoint is removed or added without having to quickly without having to train from scratch. Moreover, while not directly

applicable here, some works [6] reduce the cost of computing the leave-one-out error in the kernel regime. Furthermore, we ask if it is possible to obtain tighter bounds on the deterministic versions of the stochastic algorithms primarily considered in our work.

6.2 Statistical Methods for Data Acquisition Value

In chapter 3, we introduced the Uniform-Marginal and Independence (UMI) value function. The UMI function targets situations in which a model owner is considering acquiring (or purchasing) a dataset from a dataset owner or data vendor. The UMI function does not make any assumptions about the intended use for that function (whether it is training, evaluation or anything else). Our value function only looks at the difficulty of obtaining the dataset using the data generating model (in this case, the language model). We provided a rigorous theoretical analysis for the UMI function, and showed its clear connection to the statistical distance between the model distribution and the distribution which generated the aforementioned dataset. There are several open questions remaining relating to the data valuation in general and the UMI function specifically. We ask whether the additive property of the UMI function in **??** is really best choice, and whether a different function should be used. We also ask whether it is possible to obtain rigorous guarantees on the statistical complexity (number of samples needed) for the UMI function. Moreover, we ask whether it is possible to integrate the semantic information of the dataset into our measure.

6.3 Codes for 3D Orientation Estimation

In chapter 5, we showed that active backscatter tags can be used to great effect for 3D orientation estimation when the tag responses (codes) are designed. We proposed two design criteria which depend on the needs of the intended application (worst-case vs. average case). Our numerical simulation demonstrated the effectiveness of the proposed design criteria, and their robustness when channel knowledge is imperfect. There are multiple open questions.

We ask whether it is possible to design effective codes for time-varying environments, and whether it is possible to reduce the complexity of the optimization problem under these conditions. Moreover, it is sometimes expected the transmitters have no knowledge of the state of the channel, so an exploration of non-coherent coding is another potential future direction.

Appendix A

Omitted Details From Chapter 3

A.1 Proofs

A.1.1 Proof of Lemma 1

We begin by restating the result of Lemma 3.2.1.

Lemma. Let $\ell : \mathcal{W} \times \mathcal{Z} \to [0,1]$ be a bounded loss function, then for all $t > 0$, $w \in \mathcal{W}$, and $z^n \in \mathcal{Z}$,

$$\mathbb{E}_{u\sim\mathcal{U}}\left[\exp\left(t \cdot (\text{loo-cv}(w, z^n, u))\right)\right] \leq \exp\left(\frac{t^2 c_n^2}{8}\right), \qquad c_n = \frac{n}{n-1}. \tag{A.1}$$

Proof: To show this, let $\ell \in [0,1]^n$ and $t > 0$. For $u \in [n]$, we define:

$$f(\ell, u) = \frac{1}{n-1}\sum_{i\neq u} \ell_i - \ell_u. \tag{A.2}$$

If we obtain a bound on $\mathbb{E}_{u\sim\mathcal{U}}\left[\exp(t \cdot f(\ell, u))\right]$ for every $\ell \in [0,1]^n$, then we also obtain a bound on $\mathbb{E}_{u\sim\mathcal{U}}\left[\exp\left(t \cdot (\text{loo-cv}(w, z^n, u))\right)\right]$ for all $z^n \in \mathcal{Z}^n, w \in \mathcal{W}$. Now,

$$\mathbb{E}_{u\sim\mathcal{U}}\left[\exp(t \cdot f(\ell, u))\right] = \frac{1}{n}\exp\left(\frac{t}{n-1}\sum_{i=1}^n \ell_i\right)\sum_{j=1}^n \exp\left(-c\ell_j\right) =: \frac{1}{n}g(\ell). \tag{A.3}$$

where $c := tc_n = t(1 + \frac{1}{n-1})$. Since the maximum of g must be at a stationary point, then

$$\frac{\partial g(\ell)}{\partial \ell_j} = \frac{t}{n-1}\exp\left(\frac{t}{n-1}\sum_{i=1}^n \ell_i\right)\left(\sum_{i=1}^n e^{-c\ell_i} - ne^{-c\ell_j}\right), \tag{A.4}$$

Setting (A.4) to 0, then we must have that for all $j \in [n]$

$$\sum_{i=1}^{n} e^{-c\ell_i} - ne^{-c\ell_j} = 0. \tag{A.5}$$

Hence, for all j, we must have that $x_j \in \{0, 1, \ln(b)/c\}$, where

$$b = \frac{1}{n} \sum_{i=1}^{n} e^{-c\ell_i}. \tag{A.6}$$

Let m be the number of ℓ_j that are 0, and k the number of ℓ_j that are 1. Solving for b using (A.6), we get that

$$b = \frac{m + ke^{-c}}{m + k}. \tag{A.7}$$

We find that b depends on ℓ only through the number of 0s and 1s in ℓ. Writing $g(\ell)$ with respect to m, k and b, we obtain

$$g(\ell) = n \exp\left(\frac{t}{n-1}k + \frac{m+k}{n}\ln(b)\right), \tag{A.8}$$

Hence, finding the maximum of $g(\ell)$ is equivalent to finding the solution for the following optimization problem.

$$\begin{aligned} \max_{m,k \geq 0} \quad & \frac{t}{n-1}k + \frac{m+k}{n}\ln(b) \\ \text{s.t.} \quad & m + k \leq n. \end{aligned} \tag{A.9}$$

Since m and k are bounded non-negative integers, we can find the solution of the above maximization through a brute-force search. However, we can relax the constraints of (A.9), and allow m, k to take non-integer values over $[0, n]$. Using the method of Lagrange multipliers,

we have that:

$$\min_{\lambda\geq 0}\max_{m,k\geq 0} L(m,k,\lambda) = \min_{\lambda\geq 0}\max_{m,k\geq 0}\left(\frac{t}{n-1}k + \frac{m+k}{n}\ln(b) - \lambda(m+k-n)\right). \tag{A.10}$$

The partial derivative of (A.10) with respect to m and k are given by:

$$\nabla_m L = \frac{k(1-e^{-c})}{n(m+ke^{-c})} + \frac{\ln(b)}{n} - \lambda, \tag{A.11}$$

$$\nabla_k L = \frac{t}{n-1} - \frac{m(1-e^{-c})}{n(m+ke^{-c})} + \frac{\ln(b)}{n} - \lambda. \tag{A.12}$$

Setting the partial derivatives of (A.11) and (A.12) to 0 yields

$$m = \frac{1-e^{-c}-ce^{-c}}{c-1+e^{-c}}k \tag{A.13}$$

$$m+k = \left(\frac{c(1-e^{-c})}{c-1+e^{-c}} - 1\right)k. \tag{A.14}$$

Substituting these expressions in (A.10), we see that we now need to find

$$\begin{aligned} \max_{k\geq 0} \quad & \frac{t}{n-1}k\left(1 + \frac{(1-e^{-c})}{c-1+e^{-c}}\ln\left(\frac{1-e^{-c}}{c}\right)\right) \\ \text{s.t.} \quad & k \leq \frac{c-1+e^{-c}}{c(1-e^{-c})}n. \end{aligned}$$

Since $1 + \frac{(1-e^{-c})}{c-1+e^{-c}}\ln\left(\frac{1-e^{-c}}{c}\right) > 0$ for all $c > 0$, this expression is maximized by the largest value k can take. Hence, the we obtain our maximum for

$$\begin{aligned} k^* &= \left(-1 + \frac{c}{c(1-e^{-c})}\right)n, \\ m^* &= n-k, \\ b^* &= \frac{1-e^{-c}}{c}. \end{aligned}$$

Plugging into our expression, we have

$$g(\ell) \leq n \left(\frac{1 - e^{-c}}{c} \right) \exp \left(-1 + \frac{c}{1 - e^{-c}} \right), \tag{A.15}$$

Recalling that $\mathbb{E}_{u \sim \mathcal{U}} \left[\exp(t \cdot f(\ell, u)) \right] = 1/ng(\ell)$, we obtain

$$\mathbb{E}_{u \sim \mathcal{U}} \left[\exp(t \cdot f(\ell, u)) \right] \leq \exp \left(-1 + \frac{c}{1 - e^{-c}} + \log \left(\frac{1 - e^{-c}}{c} \right) \right) \tag{A.16}$$

Using the Taylor expansion of the exponent in the right-hand side of (A.16), we have that:

$$-1 + \frac{c}{1 - e^{-c}} + \log \left(\frac{1 - e^{-c}}{c} \right) \leq \frac{c^2}{8}. \tag{A.17}$$

Finally, we obtain

$$\mathbb{E}_{u \sim \mathcal{U}} \left[\exp \left(t \cdot (\text{loo-cv}(w, z^n, u)) \right) \right] \leq \mathbb{E}_{u \sim \mathcal{U}} \left[\exp(t \cdot f(\ell, u)) \right] \tag{A.18}$$

$$\leq \exp(\frac{c^2}{8}). \tag{A.19}$$

Recalling that $c = tc_n$, we obtain the desired result.

A.1.2 Proof of Theorem 3.2.1

We begin restating the Theorem[1].

Theorem. Let $\mathcal{A} : \mathcal{Z}^{n-1} \to \mathcal{W}$ be a training algorithm. Let $\ell : \mathcal{W} \times \mathcal{Z} \to [0, 1]$ be a bounded loss function, then

$$\text{gen}(\mathcal{A}) \leq \frac{c_n}{\sqrt{2}} \mathbb{E}_{z^n \sim \mathcal{P}^n} \sqrt{\text{loo-CMI}(\mathcal{A}, z^n)}, \tag{A.20}$$

We state a series of lemmas and definitions that are essential for the proof.

Lemma A.1.1

[1]We corrected a typo in the original paper where a $\sqrt{\cdot}$ was missing over the 2 in the denominator

Let $z^n \in \mathcal{Z}^n$, and let $w \in \mathcal{W}$, then

$$\mathbb{E}_{u\sim\mathcal{U}}\left[\text{loo-cv}(z^n, w, u)\right] = 0. \tag{A.21}$$

Proof. We have that:

$$\begin{aligned} n \cdot \mathbb{E}_{u\sim\mathcal{U}}\left[\text{loo-cv}(z^n, w, u)\right] &= \sum_{u=1}^{n}\left(\frac{1}{n-1}\sum_{i\neq u}\ell(w, z_i) - \ell(w, z_u)\right), && \text{(A.22)}\\ &= \frac{1}{n-1}\sum_{u=1}^{n}\sum_{i\neq u}\ell(w, z_i) - \sum_{u=1}^{n}\ell(w, z_u), && \text{(A.23)}\\ &= \frac{1}{n-1}\sum_{i=1}^{n}(n-1)\ell(w, z_i) - \sum_{u=1}^{n}\ell(w, z_u), && \text{(A.24)}\\ &= 0. && \text{(A.25)} \end{aligned}$$

□

Lemma A.1.2

Let $\mathcal{A} : \mathcal{Z}^{n-1} \to \mathcal{W}$ be a training algorithm, then

$$\text{gen}(\mathcal{A}) = \left|\mathbb{E}_{\mathcal{A}, z^n\sim\mathcal{P}^n, u\sim\mathcal{U}}\left[\text{loo-cv}(z^n, \mathcal{A}(z^{n-1}_{-u}), u)\right]\right| \tag{A.26}$$

Proof.

$$\begin{aligned} \text{gen}(\mathcal{A}) &= \left|\mathbb{E}_{\mathcal{A}, z^{n-1}\sim\mathcal{P}^{n-1}}\left[\mathcal{L}(\mathcal{A}(z^{n-1})) - \widehat{\mathcal{L}}(\mathcal{A}(z^{n-1}), z^{n-1})\right]\right|, && \text{(A.27)}\\ &= \left|\mathbb{E}_{\mathcal{A}, z^{n-1}\sim\mathcal{P}^{n-1}}\left[\mathbb{E}_{z'\sim\mathcal{P}}\left[\mathcal{L}(\mathcal{A}(z^{n-1}), z')\right] - \widehat{\mathcal{L}}(\mathcal{A}(z^{n-1}), z^{n-1})\right]\right|, && \text{(A.28)}\\ &= \left|\mathbb{E}_{\mathcal{A}, z^{n}\sim\mathcal{P}^{n}, u\sim\mathcal{U}}\left[\mathcal{L}(\mathcal{A}(z^{n-1}_{-u}), z_u) - \widehat{\mathcal{L}}(\mathcal{A}(z^{n-1}_{-u}), z^{n-1}_{-u})\right]\right|, && \text{(A.29)}\\ &= \left|\mathbb{E}_{\mathcal{A}, z^{n}\sim\mathcal{P}^{n}, u\sim\mathcal{U}}\left[\text{loo-cv}(z^n, u, \mathcal{A}(z^{n-1}_{-u}))\right]\right|. && \text{(A.30)} \end{aligned}$$

□

Definition A.1.1

We say that a random variable $X \sim P_X$ is σ^2-subgaussian if:

$$\mathbb{E}_{x\sim P_X}\left[t\exp(x - \mathbb{E}_{x\sim P_x}[x])\right] \leq \exp\left(\frac{t^2\sigma^2}{2}\right). \tag{A.31}$$

Lemma A.1.3 (Lemma 1, [82])

Let A, B be arbitrary random variables. Let A', B' be independent copies of A, B such that $P_{A',B'} = P_{A'}P_{B'}$. Suppose $f(A', B')$ is σ^2-subgaussian, then

$$|\mathbb{E}_{A,B}\left[g(A, B) - \mathbb{E}_{A',B'}\left[g(A', B')\right]\right]| \leq \sqrt{2\sigma^2 I(A; B)}. \tag{A.32}$$

Now, fix $Z^n = z^n$, and let $A = \mathcal{A}(z^{n-1}_{-U})$ and $B = U$, and let

$$f(A, B) = \text{loo-cv}(z^n, A, B). \tag{A.33}$$

Using Lemma A.1.1, $E_{A',B'}f(A', B') = 0$. Moreover, we use $f(A', B')$ is σ^2-subgaussian with $\sigma^2 = c_n^2/4$ using Lemma 3.2.1. This gives us that:

$$\left|\mathbb{E}_{\mathcal{A},u\sim\mathcal{U}}\left[\text{loo-cv}(z^n, \mathcal{A}(z^{n-1}_{-u}), u)\right]\right| \leq \sqrt{\frac{c_n^2}{2} I(W; U \mid z^n)}, \tag{A.34}$$

where $W = \mathcal{A}(z^{n-1}_{-U})$. Taking the expectation over z^n on both sides, we get that:

$$\mathbb{E}_{z^n\sim\mathcal{P}^n}\left|\mathbb{E}_{\mathcal{A},u\sim\mathcal{U}}\left[\text{loo-cv}(\mathcal{A}(z^{n-1}_{-u}), z^n, u)\right]\right| \leq \mathbb{E}_{z^n\sim\mathcal{P}^n}\sqrt{\frac{c_n^2}{2} I(W; U \mid z^n)}. \tag{A.35}$$

Since $g(\cdot) = |\cdot|$ is a convex function, then

$$\begin{aligned}
\text{gen}(\mathcal{A}) &= \left|\mathbb{E}_{\mathcal{A},z^n\sim\mathcal{P}^n u\sim\mathcal{U}}\left[\text{loo-cv}(\mathcal{A}(z^{n-1}_{-u}), z^n, u)\right]\right| && \text{(A.36)}\\
&\leq \mathbb{E}_{z^n\sim\mathcal{P}^n}\left|\mathbb{E}_{\mathcal{A},u\sim\mathcal{U}}\left[\text{loo-cv}(\mathcal{A}(z^{n-1}_{-u}), z^n, u)\right]\right| && \text{(A.37)}\\
&\leq \frac{c_n}{\sqrt{2}}\mathbb{E}_{z^n\sim\mathcal{P}^n}\sqrt{I(W; U \mid z^n)}. && \text{(A.38)}
\end{aligned}$$

A.1.3 Proof of Theorem 3.2.2

We begin by restating the Theorem[2].

Theorem. Let $\ell : \mathcal{R} \times \mathcal{Y} \to [0,1]$ be a bounded loss function. Let U be a uniform random variable over $[n]$, then

$$\text{gen}(h) \leq \frac{c_n}{\sqrt{2}} \mathbb{E}_{z^n \sim \mathcal{P}^n} \sqrt{\text{floo-CMI}(h, z^n)}, \tag{A.39}$$

This proof is nearly identical to the proof of Theorem 3.2.1. Here, we use the prediction function instead of the weights. We use the alternate definition of the loss $\ell : \mathcal{R} \times \mathcal{Y} \to [0,1]$. Let $g \in \mathcal{R}^n$ be a set of prediction, then we also use an alternate definition of the leave-one-out cross validation where

$$\text{loo-cv}(z^n, g, u) = \frac{1}{n-1} \sum_{i \neq u} \ell(g_i, y_i) - \ell(g_u, y_u). \tag{A.40}$$

Lemma A.1.4

Let $z^n \in \mathcal{Z}^n$, and let $g \in \mathcal{R}^n$, then

$$\mathbb{E}_{u \sim \mathcal{U}} \left[\text{loo-cv}(z^n, g, u)\right] = 0. \tag{A.41}$$

Lemma A.1.5

Let $h : \mathcal{Z}^{n-1} \times \mathcal{X} \to \mathcal{R}$ be a training algorithm, then

$$\text{gen}(\mathcal{A}) = \left| \mathbb{E}_{h, z^n \sim \mathcal{P}^n, u \sim \mathcal{U}} \left[\text{loo-cv}(z^n, h(z^{n-1}_{-u}, x^n), u)\right] \right|. \tag{A.42}$$

Now, we use Lemma A.1.3 with $A = h(z^{n-1}_{-u}, x^n)$, $B = U$, and

$$f(A, B) = \text{loo-cv}(z^n, A, B). \tag{A.43}$$

[2]We corrected a typo in the original paper where a $\sqrt{\cdot}$ was missing over the 2 in the denominator

The proof then follows in the exact same way as it did for Theorem 3.2.1, and we get the desired result:

$$\operatorname{gen}(h) \leq \frac{c_n}{\sqrt{2}} \mathbb{E}_{z^n \sim \mathcal{P}^n} \sqrt{I(h(z_{-U}^{n-1}, x^n); U \mid z^n)}. \tag{A.44}$$

A.1.4 Proof of Theorem 3.3.1

We begin by restating the Theorem.

Theorem. Let $\mathcal{A}$ be a parametric algorithm with prediction function h. Let U, Z^n, W be defined as before. Let U' be an identical independent copy of U, then

$$\text{loo-CMI}(\mathcal{A}, z^n) \leq -\mathbb{E}_{u \sim \mathcal{U}} \left[\ln \mathbb{E}_{u' \sim \mathcal{U}} \left[\exp\left(-\mathrm{KL}(p_{W \mid z^n, u} \parallel p_{W \mid z^n, u'}) \right) \right] \right], \tag{A.45}$$

$$\text{floo-CMI}(h, z^n) \leq -\mathbb{E}_{u \sim \mathcal{U}} \left[\ln \mathbb{E}_{u' \sim \mathcal{U}} \left[\exp\left(-\mathrm{KL}(p_{h(z_{-u}^{n-1}, x)} \parallel p_{h(z_{-u'}^{n-1}, x)}) \right) \right] \right]. \tag{A.46}$$

We begin by proving a more general result. Let $W \sim P_W, U \sim P_U$ be arbitrary random variables. Suppose $w \in \mathcal{W}$ and $u \in \mathcal{U}$. Let $q(u')$ be any probability distribution over $\mathcal{U}$, then we define

$$\Phi(q(u'), u) = \int_{\mathcal{W}} p(w|u) \int_{\mathcal{U}} q(u') \ln \left(\frac{p(w|u) q(u')}{p(w|u') p(u')} \right) \mathrm{d}u' \mathrm{d}w. \tag{A.47}$$

We begin with the following lemma.

Lemma A.1.6

$$I(W; U) \leq \mathbb{E}_{u \sim P_U} \left[\Phi(q, u) \right]. \tag{A.48}$$

Proof. We note that

$$\Phi(q,u) = \int_{\mathcal{W}} p(w|u) \int_{\mathcal{U}} q(u') \ln\left(\frac{p(w|u)q(u')}{p(w|u')p(u')}\right) \mathrm{d}u'\mathrm{d}w \tag{A.49}$$

$$= -\int_{\mathcal{W}} p(w|u) \int_{\mathcal{U}} q(u') \ln\left(\frac{p(w|u')p(u')}{p(w|u)q(u')}\right) \mathrm{d}u'\mathrm{d}w. \tag{A.50}$$

$$= -\int_{\mathcal{W}} p(w|u) \mathbb{E}_{u'\sim q_U}\left[\ln\left(\frac{p(w|u')p(u')}{p(w|u)q(u')}\right)\right] \mathrm{d}w. \tag{A.51}$$

$$\geq -\int_{\mathcal{W}} p(w|u) \ln\left(\mathbb{E}_{u'\sim q_U}\left[\frac{p(w|u')p(u')}{p(w|u)q(u')}\right]\right) \mathrm{d}w, \tag{A.52}$$

where the last inequality follows from the concavity of ln and Jensen's inequality. Continuing from (A.52), we obtain

$$\Phi(q,u) \geq -\int_{\mathcal{W}} p(w|u) \ln\left(\mathbb{E}_{u'\sim P_U}\left[\frac{p(w|u')p(u')}{p(w|u)q(u')}\right]\right) \mathrm{d}w \tag{A.53}$$

$$= -\int_{\mathcal{W}} p(w|u) \ln\left(\int_{\mathcal{U}} q(u')\frac{p(w|u')p(u')}{p(w|u)q(u')}\mathrm{d}u'\right) \mathrm{d}w \tag{A.54}$$

$$= -\int_{\mathcal{W}} p(w|u) \ln\left(\int_{\mathcal{U}} \frac{p(w|u')p(u')}{p(w|u)}\mathrm{d}u'\right) \mathrm{d}w \tag{A.55}$$

$$= -\int_{\mathcal{W}} p(w|u) \ln\left(\frac{p(w)}{p(w|u)}\right) \mathrm{d}w \tag{A.56}$$

$$= \mathrm{KL}(p(w|u) \parallel p(w)). \tag{A.57}$$

Since $I(W;U) = \mathbb{E}_{u\sim P_U}\left[\mathrm{KL}\left(p(w|u)\|\|p(w)\right)\right]$, we obtain the result of the lemma. □

To find a tight bound, we minimize $\Phi(q(u'),u)$ with respect to the probability distribution $q(u')$. Since $\phi(q(u',u)$ is an upper bound to $\mathrm{KL}\left(p(w|u)\|\|p(w)\right)$ for any $q(u')$, the infimum of Φ with respect to $q(u')$ is still an upper bound. Since we are trying to find the infimum with respect to a probability distribution, the Lagrangian is given by

$$\tilde{\Phi}(q(u')) = \int_{\mathcal{W}} p(w|u) \int_{\mathcal{U}} q(u') \ln\left(\frac{p(w|u)q(u')}{p(w|u')p(u')}\right) \mathrm{d}u'\mathrm{d}w + \lambda\left(\int_{\mathcal{U}} q(u')\mathrm{d}u' - 1\right). \tag{A.58}$$

We "differentiate" with respect to $q(u')$, and obtain

$$\frac{\partial\tilde{\Phi}(q(u'))}{\partial q(u')} = \int_{\mathcal{W}} p(w|u)\ln\left(\frac{p(w|u)q(u')}{p(w|u')p(u')}\right)\mathrm{d}w + 1 + \lambda. \tag{A.59}$$

Setting (A.59) to 0, and solving for $q(u')$ for every $u' \in \mathcal{U}$, we obtain

$$q^*(u') = \frac{p(u')e^{-\mathrm{KL}(p(w|u)\|\|p(w|u'))}}{\int_{\mathcal{U}} p(\hat{u})e^{-\mathrm{KL}(p(w|u)\|\|p(w|\hat{u}))}\mathrm{d}\hat{u}} \tag{A.60}$$

To find the expression for $\Phi(q^*(u'), u)$, note that:

$$\mathrm{KL}(q^*(u') \parallel p(u')) = -\ln\left(\int_{\mathcal{U}} p(u')e^{-\mathrm{KL}(p(w|u) \parallel p(w|u'))}\mathrm{d}u'\right) - \int_{\mathcal{U}} q^*(u')\mathrm{KL}(p(w|u) \parallel p(w|u'))\mathrm{d}u'. \tag{A.61}$$

Plugging in the above into $\Phi(q^*(u'), u)$, we obtain

$$\begin{aligned}
\Phi(q^*(u'), u) &= \int_{\mathcal{U}} q^*(u')\mathrm{KL}(p(w|u) \parallel p(w|u'))\mathrm{d}u' + \mathrm{KL}(q^*(u') \parallel p(u')) && \text{(A.62)}\\
&= -\ln\left(\int_{\mathcal{U}} p(u')e^{-\mathrm{KL}(p(w|u) \parallel p(w|u'))}\mathrm{d}u'\right) && \text{(A.63)}\\
&= -\ln \mathbb{E}_{u'\sim P_U}\left[e^{-\mathrm{KL}(p(w|u) \parallel p(w|u'))}\right] && \text{(A.64)}
\end{aligned}$$

Finally, taking the expectation with respect to U and using Lemma A.1.6, we obtain that

$$I(W;U) \le -\mathbb{E}_{u\sim P_U}\left[-\ln \mathbb{E}_{u'\sim P_U}\left[\mathrm{KL}(p_{W\,|\,u}) \parallel p_{W\,|\,u'})\right]\right]. \tag{A.65}$$

We apply the above to $I(W;U|z^n)$ and $I(h(z_{-i}^{n-1}, x^n), U \mid z^n)$ to obtain the expressions of Theorem 3.3.1.

A.1.5 Proof of Theorem 3.3.2

We begin a restatement of the theorem and the following lemma:

Theorem. Let $\mathcal{A} : \mathcal{Z}^{n-1} \to \mathcal{W}$ be a deterministic algorithm, and let $\ell : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}_+$ be the

loss that a set of weights incur on a sample. If $\ell(w, \cdot)$ is L-Lipschitz in the weights, then if $\mathcal{A}$ has ϵ-weight stability relative to a positive semi-definite Σ, then $\text{gen}(\mathcal{A}) \leq \sqrt{4c_n \epsilon L \sqrt{\text{tr}(\Sigma)}}$.

Lemma A.1.7

Let $\mathcal{A} : \mathcal{Z}^{n-1} \to \mathcal{W}$ be a deterministic algorithm Let the loss $\ell : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}_+$ be L-Lipschitz in the weights. Let Σ be a positive definite matrix, then

$$\text{gen}(\mathcal{A}) \leq \text{gen}(\mathcal{A}_\Sigma) + 2L\sqrt{\text{tr}(\Sigma)}. \tag{A.66}$$

Proof. We have that:

$$\begin{aligned}
\text{gen}(\mathcal{A}) &= \left| \mathbb{E}_{\mathcal{A}, z^n \sim \mathcal{P}^n} \left[\mathcal{L}(\mathcal{A}(z^n)) - \widehat{\mathcal{L}}(\mathcal{A}(z^n), z^n) \right] \right|. && \text{(A.67)} \\
&= \left| \mathbb{E}_{\mathcal{A}, z^n \sim \mathcal{P}^n, z' \in \mathcal{P}} \left[\ell(\mathcal{A}(z^n), z') - \frac{1}{n} \sum_{i=1}^n \ell(\mathcal{A}(z^n), z_i) \right] \right|. && \text{(A.68)} \\
&= \left| \mathbb{E}_{\mathcal{A}, z^n \sim \mathcal{P}^n, z' \in \mathcal{P}} \left[\ell(\mathcal{A}_\Sigma(z^n), z') + \delta' - \frac{1}{n} \sum_{i=1}^n \left(\ell(\mathcal{A}_\Sigma(z^n), z_i) + \delta_i \right) \right] \right|, && \text{(A.69)} \\
&\leq \left| \mathbb{E}_{\mathcal{A}, z^n \sim \mathcal{P}^n, z' \in \mathcal{P}} \left[\ell(\mathcal{A}_\Sigma(z^n), z') - \frac{1}{n} \sum_{i=1}^n \left(\ell(\mathcal{A}(z^n), z_i) \right) \right] \right| + |\delta'| + \frac{1}{n} \sum_{i=1}^n |\delta_i|, && \text{(A.70)} \\
&\leq \text{gen}(\mathcal{A}_\Sigma) + \mathbb{E}_{\mathcal{A}, z^n \sim \mathcal{P}^n, z' \in \mathcal{P}} \left[|\delta'| + \frac{1}{n} \sum_{i=1}^n |\delta_i| \right], && \text{(A.71)}
\end{aligned}$$

where $\delta' = \ell(\mathcal{A}(z^n), z') - \ell(\mathcal{A}_\Sigma(z^n), z')$ and $\delta_i = \ell(\mathcal{A}(z^n), z_i) - \ell(\mathcal{A}_\Sigma(z^n), z_i)$. Recalling that $\mathcal{A}_\Sigma(z^n) = \mathcal{A}(z^n) + N$ where $N \sim \mathcal{N}(0, \Sigma)$, we have that $|\delta'| \leq L \|N\|$ (also for $|\delta_i|$). This gives us that:

$$\begin{aligned}
\text{gen}(\mathcal{A}) &\leq \text{gen}(\mathcal{A}_\Sigma) + \mathbb{E}_{\mathcal{A}, z^n \sim \mathcal{P}^n, z' \in \mathcal{P}} \left[|\delta'| + \frac{1}{n} \sum_{i=1}^n |\delta_i| \right], && \text{(A.72)} \\
&= \text{gen}(\mathcal{A}_\Sigma) + 2L \|N\|, && \text{(A.73)} \\
&\leq \text{gen}(\mathcal{A}_\Sigma) + 2L\sqrt{\text{tr}(\Sigma)}, && \text{(A.74)}
\end{aligned}$$

where the last inequality follows from the fact that for $N \sim \mathcal{N}(0, \Sigma)$, $\mathbb{E}[\|N\|] \leq \text{tr}(\Sigma)$. □

Now, suppose $\mathcal{A}$ has ϵ-weight stability relative to Σ, and let $\alpha > 0$, then using (3.8) and (3.16), we have

$$\text{gen}(\mathcal{A}_{\alpha^2\Sigma}) \leq \frac{c_n \epsilon}{2\alpha}. \tag{A.75}$$

Hence, for all $\alpha > 0$, we have that:

$$\text{gen}(\mathcal{A}) \leq \frac{c_n \epsilon}{2\alpha} + 2L\alpha\sqrt{\text{tr}(\Sigma)}. \tag{A.76}$$

Finding the minimum of the above with respect to α, we obtain:

$$\text{gen}(\mathcal{A}) \leq 2\sqrt{\frac{c_n \epsilon}{2} \cdot 2L\sqrt{\text{tr}(\Sigma)}}, \tag{A.77}$$

$$= \sqrt{4c_n \epsilon L\sqrt{\text{tr}(\Sigma)}}. \tag{A.78}$$

A.1.6 Proof of Theorem A.1.6

We begin with a restatement of the theorem, and the following lemma where proof is identical to the lemma used in Section A.1.5.

Theorem. Let $h : \mathcal{Z}^n \times \mathcal{X} \to \mathbb{R}^d$ be a deterministic prediction function. Assume that h has β-train stability and β_1-test stability. Assume the loss function $\ell : \mathcal{R} \times \mathcal{Y}$ is L-Lipschitz continuity in the first coordinate, then $\text{gen}(\mathcal{A}) \leq \sqrt{4c_n L\sqrt{nd(n\beta^2 + 2\beta_1^2)}}$.

Lemma A.1.8

Let $h : \mathcal{Z}^{n-1} \times \mathcal{X} \to \mathcal{R} \subset \mathbb{R}^d$ be a deterministic prediction function. Let the loss $\ell : \mathcal{R} \times \mathcal{Y} \to \mathbb{R}_+$ be L-Lipschitz in the predictions. Let σ^2 be a positive definite matrix, then

$$\text{gen}(h) \leq \text{gen}(h_\sigma) + 2L\sigma\sqrt{nd}. \tag{A.79}$$

Now, for a prediction function h with β-train stability and β_1-test stability, we have that

$$\text{gen}(h_\sigma) \leq \frac{c_n\sqrt{(n-2)\beta^2 + 2\beta_1^2}}{2\sigma} \leq \frac{c_n\sqrt{n\beta^2 + 2\beta_1^2}}{2\sigma}. \tag{A.80}$$

Hence, we have that

$$\text{gen}(h) \leq \frac{c_n\sqrt{n\beta^2 + 2\beta_1^2}}{2\sigma} + 2L\sigma\sqrt{nd}. \tag{A.81}$$

Optimizing the right-hand-side of the above over σ, we obtain the desired result:

$$\text{gen}(h) \leq \sqrt{4c_nL\sqrt{nd(n\beta_1 + 2\beta_1^2)}}. \tag{A.82}$$

A.1.7 Proof of Lemma A.1.7

We restate the lemma[3], and we provide a definition and a lemma used in the proof.

Lemma. Let the gradient update rule be γ-bounded. Suppose we run SGD for T steps, and we use the same starting value for the weights, then $\text{gen}(\mathcal{A}_{\sigma^2 I}) \leq \sqrt{\frac{c_n^2 T^2 \gamma}{\sigma^2}}$.

Definition A.1.2 (Definition 2.4, [28])
An update step G is said to γ-bounded if for all $w \in \mathcal{W}$,

$$\|G(w) - w\| \leq \sqrt{\gamma}. \tag{A.83}$$

Lemma A.1.9 (Lemma 2.5, [28])
Let $G_1, \ldots, G_T$ and $G'_1, \ldots, G'_T$ be two arbitrary sequences of updates. Let $w_0 = w'_0$ (same initial weights), and let $\delta_t = \|w_t - w'_t\|$ where w_t and w'_t are defined recursively through $w_{t+1} = G_t(w_t)$ and $w'_{t+1} = G'_t(w'_t)$. If G_t and G'_t are γ-bounded, then

$$\delta_{t+1} \leq \delta_t + 2\sqrt{\gamma}. \tag{A.84}$$

Let $G_1, \ldots, G_T$ be the sequence of updates made when SGD is used with the dataset z_{-i}^{n-1}, and let $G'_1, \ldots, G'_T$ be the sequence of updates made when SGD is used with the dataset z_{-j}^{n-1}. If the previous updates are γ-bounded, then Lemma A.1.9 gives us a bound on $\|w_{-i} - w_{-j}\|$ if we use T iteration of SGD, and we obtain the desired result.

[3]We corrected a typo in the original paper where a 2 was a denominator instead of an exponent for T.

A.2 Additional Experiments

We fine-tune an ImageNet pre-trained ResNet-18 model on MIT67, Oxford Flowers and Oxford Pets datasets. In order to compute the information bounds we use the results in corollary 3.3.1. We compute w_{-i}/h_{-i} by removing sample i from the training set and re-training from scratch. For all the experiments we remove 10 samples one at a time across 3 random seeds and use corollary 3.3.1 to compute the information bounds. We used 2 NVIDIA 1080Ti GPUS and the experiments take 1-2 days.

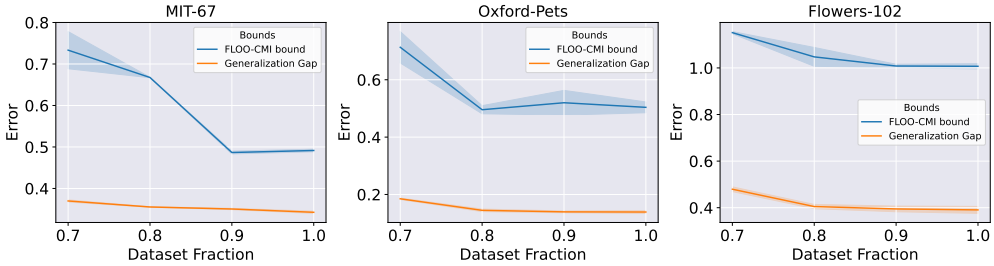

Figure A.1: We repeat the same experiment as in fig. 3.2 but with a larger size for the number of samples to estimate the information bounds

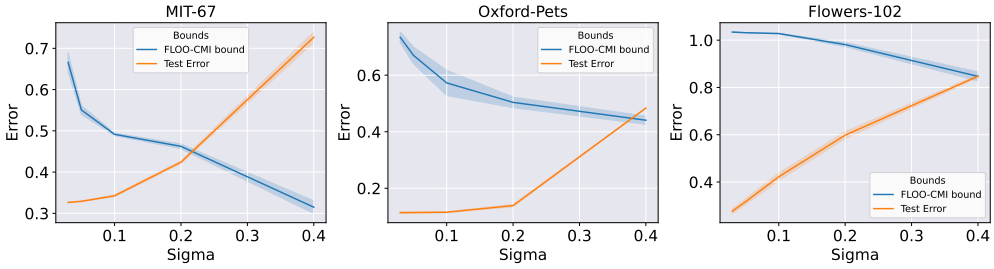

Figure A.2: We repeat the same experiment as in fig. 3.3 but with a larger size for the number of samples to estimate the information bounds

Appendix B

Omitted Details From Chapter 4

B.1 Proofs

B.1.1 Univariate Probability Integral Transform

We include the following theorem and its proof for completeness.

Theorem. Suppose X is a real-valued random variable with a continuous distribution (cdf) F_X, then the random variable defined as $Y := F_X(X)$ is uniformly distributed on the range $(0, 1)$.

Proof. We define the generalized inverse of the cdf F_X as

$$F_X^{-1}(y) = \inf\{x : F_X(x) \geq y\}.$$

When F_X is strictly increasing, the generalized inverse coincides with the standard definition of the inverse i.e. $F_X^{-1}(y) = x \iff F_X(x) = y$. However, if F_X is constant on some interval, then the standard definition fails. The generalized inverse takes care of this by assigning to $F_X^{-1}(y)$ a single value. Now, let $0 < y < 1$, then

$$\Pr(Y \leq y) = \Pr(F_X(X) \leq y) \tag{B.1}$$
$$= \Pr(F_X^{-1} F_X(X) \leq F_X^{-1}(y)) \tag{B.2}$$
$$= \Pr(X \leq F_X^{-1}(y)) \tag{B.3}$$
$$= F_X\left(F_X^{-1}(y)\right) \tag{B.4}$$
$$= y. \tag{B.5}$$

The justification behind the second line second line is F_X^{-1} is always strictly increasing. The justification behind the third line is a bit tricky. If F_X is strictly increasing, then $F_X^{-1} F_X(x) = x$. However, if F_X is constant on some interval $I = [a, b]$, then $F_X^{-1} F_X(x) \neq x$ for all $x \in I \setminus \{a\}$. But since $P(X \leq x) = P(X \leq a)$ for all $x \in I$, this ends up making no difference. □

B.1.2 Proof of Theorem 4.3.2

Proof. The proof uses two properties of f-divergences which we state and prove.

Lemma B.1.1

Let X and Y be any two random variables. Let $P_{X,Y} = P_X P_{Y|X}$ and $Q_{X,Y} = Q_X Q_{Y|X}$ be any two joint distributions, then

$$D_f\left(P_{X,Y} \,||\, Q_{X,Y}\right) \geq D_f\left(P_X \,||\, Q_X\right). \tag{B.6}$$

Proof. We have

$$D_f\left(P_{X,Y} \mid\mid Q_{X,Y}\right) = E_{X\sim Q_X} E_{Y\sim Q_{Y|X}}\left[f\left(\frac{\mathrm{d}P_X P_{Y|X}}{\mathrm{d}Q_X Q_{Y|X}}\right)\right] \tag{B.7}$$

$$\geq E_{X\sim Q_X}\left[f\left(E_{Y\sim Q_{Y|X}}\frac{\mathrm{d}P_X P_{Y|X}}{\mathrm{d}Q_X Q_{Y|X}}\right)\right] \tag{B.8}$$

$$= E_{X\sim Q_X}\left[f\left(\frac{\mathrm{d}P_X}{\mathrm{d}Q_X}\right)\right] \tag{B.9}$$

$$= D_f(P_X \mid\mid Q_X) \tag{B.10}$$

We use the convexity of f and Jensen's inequality in Eq. B.8. In Eq. B.9, we use

$$E_{Y\sim Q_{Y|X}}\left[\frac{\mathrm{d}P_X P_{Y|X}}{\mathrm{d}Q_X Q_{Y|X}}\right] = \int \frac{p_X(x)p_{Y|X}(y)}{q_X(x)q_{Y|X}(y)} Q_{Y|X}(\mathrm{d}y) \tag{B.11}$$

$$= \frac{p_X(x)}{q_X(x)}\int \frac{p_{Y|X}(y)}{q_{Y|X}(y)} q_{Y|X}(y)\mu(\mathrm{d}y) \tag{B.12}$$

$$= \frac{p_X(x)}{q_X(x)}\int p_{Y|X}(y)\mu(\mathrm{d}y) \tag{B.13}$$

$$= \frac{p_X(x)}{q_X(x)}. \tag{B.14}$$

□

Lemma B.1.2

Consider a channel that produces Y given X based on the conditional law $P_{Y|X}$. Let P_Y and Q_Y denote the distributions of Y when X is distributed as P_X and Q_X respectively, then for any f-divergence, we have

$$D_f(P_Y \mid\mid Q_Y) \leq D_f(P_X \mid\mid Q_X). \tag{B.15}$$

Proof. Let $P_{X,Y} = P_X P_{Y|X}$ and $Q_{X,Y} = Q_X P_{Y|X}$. Since the two joint distributions share the same conditional law, we get that $D_f\left(P_X \mid\mid Q_X\right) = D_f\left(P_{X,Y} \mid\mid Q_{X,Y}\right)$. Now, we use Lemma B.1.1 to obtain the result. □

We obtain Eq. 4.8 by applying Lemma B.1.2 twice. Once with X as the token and $Y = \tilde{F}_p(\tilde{X})$ which gives $D_f(G\|\mathcal{U}) \leq D_f(Q\|P)$, and another time with the roles of X and Y reversed to obtain $D_f(Q\|P) \geq D_f(G\|\mathcal{U})$. Note that we can reverse the roles of X and Y as we can recover the original token through $\left\lceil \tilde{F}_p^{-1}(\tilde{F}_p(\tilde{X})) \right\rceil$. Combining the two inequalities, we obtain the desired result. □

B.1.3 Proof of Theorem 4.3.3

Proof. Since Markov chain $\{X_{q,t}\}$ converges, we know that for any $x_{1:m} \in \mathcal{V}^m$, $\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} \mathbf{1}(x_{1:m} = X_{q,\tau+1:\tau+m}) = q^{\infty}(x_{1:m})$ almost surely. It thus follows that

$$\begin{aligned}
\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} G_\tau(\cdot) &= \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} F_{Y_p|m}(\cdot|X_{q,\tau+1:\tau+m}) \\
&= \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} \sum_{x_{1:m}\in\mathcal{V}^m} \sum_{\tau=1}^{t} F_{Y_p|m}(\cdot|x_{1:m})\mathbf{1}(x_{1:m} = X_{q,\tau+1:\tau+m}) \\
&= \sum_{x_{1:m}\in\mathcal{V}^m} F_{Y_p|m}(\cdot|x_{1:m}) \left(\frac{1}{t}\sum_{\tau=1}^{t} \mathbf{1}(x_{1:m} = X_{q,\tau+1:\tau+m})\right).
\end{aligned}$$

We then have $\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} G_\tau(\cdot) = \sum_{x_{1:m}\in\mathcal{V}^m} q^{\infty}(x_{1:m}) F_{Y_p|m}(\cdot|x_{1:m})$.

$$\begin{aligned}
D_f\left(\sum_{x_{1:m}\in\mathcal{V}^m} q^{\infty}(x_{1:m}) F_{Y_p|m}(\cdot|x_{1:m})||\mathcal{U}\right) &\leq \sum_{x_{1:m}\in\mathcal{V}^m} q^{\infty}(x_{1:m}) D_f\left(F_{Y_p|m}(\cdot|x_{1:m})||\mathcal{U}\right) \\
&= \sum_{x_{1:m}\in\mathcal{V}^m} q^{\infty}(x_{1:m}) D_f(Q(\cdot|x_{1:m})||P(\cdot|x_{1:m})),
\end{aligned}$$

where the first relation is by Jenson's inequality and the second relation is calling the equality from the memory-less case. □

B.1.4 Proof of Transformation Change

Lemma B.1.3

Let $x \in [k]$, $u \in (0,1)$, $y = x - u$, $p(\cdot \mid x_{1:i-1})$ a probability mass function over $[k]$, and

$\tilde{F}(\cdot \mid x_{1:i-1})$ *be defined as in 4.6, then*

$$\tilde{F}(y \mid x_{1:i-1}) = up(x \mid x_{1:i-1}) + F(x \mid x_{1:i-1}). \tag{B.16}$$

Proof. Using the definition of $\tilde{F}(y_i \mid x_{1:i-1})$ and the fact that $\lfloor y \rfloor = x - 1$ and $\lceil y \rceil = x$, we have

$$\tilde{F}(y \mid x_{1:i-1}) = \sum_{j=1}^{\lfloor y \rfloor} p(j \mid x_{1:i-1}) + (y - \lfloor y \rfloor) p(\lceil y \rceil \mid x_{1:i-1}) \tag{B.17}$$

$$= F(x - 1 \mid x_{1:i-1}) + up(x \mid x_{1:i-1}). \tag{B.18}$$

This proves our assertion. □

B.1.5 Multivariate PIT

To extend this theorem to the case of multivariate random variables, a straightforward application of the multivariate cumulative distribution function will not work. To see why, consider the continuous random vector $X = (X_1, X_2)$. Let F_i be the marginal distribution function of X_i for $i = 1, 2$, and F the joint distribution function. Let x_1, x_2 be any two real numbers, then

$$F(x_1, x_2) = \Pr(X_1 \leq x_1, X_2 \leq x_2), \tag{B.19}$$

$$= \Pr(F(X_1) \leq x_1, F(X_2) \leq F(x_2)), \tag{B.20}$$

$$= C(F(x_1), F(x_2)) \tag{B.21}$$

where $C(u_1, u_2)$ is the distribution function of $(U_1, U_2) = (F_1(X_1), F_2(X_2))$. C is referred to as a copula in probability theory because it joins or "couples" the joint distribution function F to its marginal distribution functions F_1, F_2. While both U_1 and U_2 have a marginal uniform distribution, U_1 and U_2 are generally not independent, and their joint distribution rests entirely on the joint distribution of X_1 and X_2. For instance, if $X = (Z, 2Z)$ for

some continuous random variable Z, then the corresponding copula can be shown to be $C(u_1, u_2) = \min\{u_1, u_2\}$.

B.1.6 Estimating the Kullback-Leibler

B.2 Statistical Tests

B.2.1 Testing Uniformity

B.2.1.1 Kolmogorov-Smirnov

Given n observations $U_1, \cdots, U_n$, we find first compute the empirical cumulative distribution function

$$F_n(X) = \frac{1}{n} \sum_{j=1}^{n} 1_{U_i \leq x} \tag{B.22}$$

The Kolmogorov–Smirnov statistic for a given cumulative distribution function $F(x)$ is then

$$D_n = \sup_x |F_n(x) - F(x)| \tag{B.23}$$

The Wiener process W_t is the characterised by the following properties:

- $W_0 = 0$ almost surely
- W has independent increments: for every $t > 0$, $W_{t+u} - W_t, u \geq 0$ are independent of the past values $W_s, s < t$
- W has Gaussian increments: $W_{t+u} - W_t$ is normally distributed with mean 0 and variance u i.e. $W_{t+u} - W_t \sim \mathcal{N}(0, u)$.
- W has almost surely continuous paths: W_t is almost surely continuous in t.

Define the Brownian bridge as the stochastic process whose probability distribution is the conditional probability distribution of a standard weiner process subject to $W_1 = 0$, so that the

process is pinned to the same value at both $t = 0$ and $t = 1$ i.e. $B_t := (W_t \mid W_1 = 0),\ t \in [0, 1]$ or in other words:

$$B_t = W_t - tW_1 \tag{B.24}$$

Define the random variable $K = \sup_{t\in[0,1]}|B(t)|$. This random variable has what is called the Kolmogorov distribution:

$$\Pr(K \leq x) = 1 - 2\sum_{k=1}^{\infty}(-1)^{k-1}e^{-2k^2x^2} = \frac{\sqrt{2\pi}}{x}\sum_{k=1}^{\infty}e^{-(2k-1)^2\pi^2/(8x^2)} \tag{B.25}$$

Theorem B.2.1

Under null hypothesis that the sample comes from the hypothesized continuous distribution $F(x)$,

$$\sqrt{n}D_n \xrightarrow[d]{n\to\infty} K \tag{B.26}$$

The goodness-of-fit test or the Kolmogorov–Smirnov test can be constructed by using the critical values of the Kolmogorov distribution. This test is asymptotically valid when $n \to \infty$. It rejects the null hypothesis at level α if

$$\sqrt{n}D_n > K_\alpha, \tag{B.27}$$

where K_α is found from $\Pr(K \leq K_\alpha) = 1 - \alpha$.

B.2.1.2 Chi-Squared Test

Let $[p_1, \ldots, p_k]$ be a discrete distribution over the categories $\{1, \ldots, k\}$. Let $Y_1, \ldots, Y_n$ be n samples from a discrete distribution taking values over $\{1, \ldots, k\}$. Let $(O_1, O_2, ..., O_n)$ be the count number of samples from a finite set of given categories. They satisfy $\sum_{i=1}^{k} O_i = n$.

The null hypothesis is that the count numbers are sampled from a multinomial distribution Multinomial$(n; p_1, ..., p_k)$. That is, the underlying data is sampled IID from a categorical distribution Categorical$(p_1, ..., p_n)$ over the given categories.The Pearson's chi-squared test

statistic is defined as

$$V := \sum_i \frac{(O_i - np_i)^2}{np_i} \tag{B.28}$$

Theorem B.2.2

Under the null hypothesis $V \sim \chi^2_{k-1}$ i.e. a chi-squared distribution with $\nu = k - 1$ degrees of freedom.

It is important to note that the chi-squared test is designed for discrete random variables. Let d be some positive integer, and define the new sequence $Y_1, \ldots, Y_n$ as

$$Y_j = \lfloor dU_j \rfloor \tag{B.29}$$

This is a sequence of integers that purports to be independently and uniformly distributed between 0 and $d - 1$. The number d is chosen for convenience; for example, we might have $d = 64 = 2^6$ on a binary computer, so that Y_n represents the six most significant bits of the binary representation of U_n. The value of d should be large enough so that the test is meaningful, but not so large that the test becomes impracticably difficult to carry out.

B.2.2 Independence Testing

B.2.2.1 Poker test (Partition test)

The "classical" poker test considers n groups of five successive integers, $\{Y_{5j}, Y_{5j+1}, \ldots, Y_{5j+4}\}$ for $0 \leq j < n$, and observes which of the following seven patterns is matched by each (orderless) quintuple:

- All different: abcde
- One pair: aabcd
- Two pairs: aabbc

- Three of a kind: aaabc
- Full house: aaabb
- Four of a kind: aaaab
- Five of a kind: aaaaa

A chi-square test is based on the number of quintuples in each category. A simpler version of this test which is almost just as good is to count a number of distinct values in the set of five. In which case,

- 5 values = all different;
- 4 values = one pair;
- 3 values = two pairs, or three of a kind;
- 2 values = full house, or four of a kind;
- 1 value = five of a kind.

This breakdown is easier to determine systematically. In general, we can consider n groups of k successive numbers, and we can count the number of k-tuples with r different values. A chi-square test is then made, using the probability

$$p_r = \frac{d(d-1)\cdots(d-r+1)}{d^k} \left\{ {k \atop r} \right\}, \tag{B.30}$$

where $\left\{ {k \atop r} \right\}$ is the stirling number of the second kind. The value of p_r is usually very low for $r = 1$ and $r = 2$, so we usually lump those together before applying the chi-squared test.

B.2.2.2 Permutation Test

Divide the input sequence into n groups of t elements each i.e. $V_j = \max(U_{tj}, U_{tj+1}, \ldots, U_{tj+t-1})$ for $0 \leq j < n$. The elements in each group can have $t!$ possible relative orderings; the number

of times each ordering appears is counted, and a chi-square test is applied with $k = t!$ and with probability $1/t!$ for each ordering.

In order to perform this test, we need a method of indexing permutations. The set $\{0, 1, 2\}$ can be permuted $3! = 6$ ways. Those 6 permutations and their lexicographic ranks are:

- $0 \to (0, 1, 2)$
- $1 \to (0, 2, 1)$
- $2 \to (1, 0, 2)$
- $3 \to (1, 2, 0)$
- $4 \to (2, 0, 1)$
- $5 \to (2, 1, 0)$

Calculating sequential indexes for permutations is done by computing the Lehmer code of the permutation, and then converting that **Lehmer code** to a base-10 number. Like that base-10 number, a Lehmer code is just a sequence of digits; However, each digit has a different base. Technically, it's a mixed-radix numeral system known as a factorial number system.

First, let σ be a permutation of n numbers $\{0, \ldots, n-1\}$. For example, if $n = 3$, we can represent σ as

$$\sigma = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \end{pmatrix} \tag{B.31}$$

This means that σ places in the 0th position what used to be in the 1st position, in the 1st position what used to be in the 2nd position, and in the 2nd position what used to be in the 0th position. So $\sigma((0, 1, 2)) = (1, 2, 0)$ and $\sigma((2, 0, 1)) = (0, 1, 2)$. A more compact notation of the above is $\sigma = (\sigma_0, \ldots, \sigma_{n-1})$ which just include the second row of the above representation. The Lehmer code of a permutation $L(\sigma) = (L(\sigma)_0, \ldots, L(\sigma)_{n-1})$ is given by:

$$L(\sigma)_i = |\{j > i : \sigma_j < \sigma_i\}| \tag{B.32}$$

In other words, $L(\sigma)_i$ counts the number of terms to the right of σ_i in $(\sigma_0, \ldots, \sigma_{n-1})$ which are smaller than it, and the number is between 0 and $n-i-1$; A pair of indices (i,j) with $i<j$ and $\sigma_i > \sigma_j$ is called an inversion of σ, and $L(\sigma)_i$ counts the number of inversions (i,j) with i fixed and varying j. For example, $L((1,2,0)) = (1,1,0)$. Moreover, we can easily convert the Lehmer code into the permutation index. The index of the permutation is then $\sum_{i=0}^{n-1} L(\sigma)_i (n-i-1)!$ e.g. the index of $(1,2,0)$ is $1(2!)+1(1!)+0(0!) = 3$.

We can find the Lehmer code in linear time. Let $b = 00\cdots 0b$ be a binary number with n bits and let $\sigma = (\sigma_0, \ldots, \sigma_{n-1})$ be our permutation, then for $i = 0, \ldots, n-1$, we

1. We flip bit the σ_ith bit of b
2. Right-Shift b by $n - \sigma_i$ to get b'.
3. Count the number of 1s in b' and subtract it from σ_i to get the Lehmer code $L(\sigma)_i$.

Proof of correctness: Say we are at position i in the loop. Until now bit j of b is 1 iff j is to the left of σ_i. We you shift b by $n - \sigma_i$ positions to the right, and what would be left is the first σ_i bits of b corresponding to the numbers $0, \ldots, \sigma_i - 1$.

B.2.2.3 Serial Correlation Test

We may also compute the following statistic:

$$C = \frac{n\left(U_0U_1 + U_1U_2 + \ldots U_{n-2}U_{n-1} + U_{n-1}U_0\right) - \left(U_0 + U_1 + \ldots + U_{n-1}\right)^2}{n\left(U_0^2 + U_1^2 + \ldots + U_{n-1}^2\right) - \left(U_0 + U_1 + \ldots + U_{n-1}\right)^2}. \tag{B.33}$$

This is the "serial correlation coefficient," a measure of the extent to which U_{j+1} depends on U_j. Correlation coefficients appear frequently in statistical work. If we have n quantities $U_0, U_1, \ldots, U_{n-1}$ and n others $V_0, V_1, \ldots, V_{n-1}$, coefficient between them is defined to be:

$$C = \frac{n\sum(U_jV_j) - (\sum U_j)(\sum V_j)}{\sqrt{\left(n\sum U_j^2 - (\sum U_j)^2\right)\left(n\sum V_j^2 - (\sum V_j)^2\right)}} \tag{B.34}$$

When C is zero or very small, it indicates that the quantities U_j and V_j are uncorrelated (a weaker notion than independence), whereas a value of ± 1 indicates total linear dependence. Therefore it is desirable to have C close to zero. In actual fact, since U_0U_1 is not completely independent of U_1U_2, the serial correlation coefficient is not expected to be exactly zero.

In other words, we are finding the correlation between $(U_0, U_1, \ldots, U_{n-1})$ and $(U_1, \ldots, U_{n-1}, U_0)$. We can also compute the correlation coefficient between $(U_0, U_1, \ldots, U_{n-1})$ and any cyclically shifted sequence $(U_q, \ldots, U_{n-1}, U_0, \ldots, U_{q-1})$. A naive approach to computing the cyclic correlations takes $\mathcal{O}(n^2)$. However, one can use the Fast Fourier Transform to find all these correlations in $\mathcal{O}(n \log n)$.

B.2.2.4 Serial Test

More generally, we want pairs of successive numbers to be uniformly distributed in an independent manner. We can also use the chi-squared test to test for independence (somewhat).

To carry out the serial test, we simply count the number of times that the pair $(Y_{2j}, Y_{2j+1}) = (q, r)$ occurs, for $0 <= j < n$; these counts are to be made for each pair of integers (q, r) with $0 \leq q, r < d$, and the chi-square test is applied to these $k = d^2$ categories with probability $1/d^2$ in each category.

As with the equidistribution test, d may be any convenient number, but it will be somewhat smaller than the values suggested above since a valid chi-square test should have n large compared to k. A common rule of thumb is to take n large enough so that each of the expected values $np_i \geq 5$ i.e. the expected number of occurrences of each categorical outcomes is five or more; preferably, however, take n much larger than this, to get a more powerful test. Hence, here we need $n \geq 5d^2$.

B.2.2.5 Gap Test

Another test is used to examine the length of "gaps" between occurrences of U_j in a certain range. Let α and β be two real numbers with $0 \leq \alpha < \beta \leq 1$, and suppose $U_j \in [\alpha, \beta)$, then

we want to consider the lengths of consecutive subsequences $U_j, U_{j+1}, ..., U_{j+r}$ in which U_{j+r} lies between α and β but the other U's do not. This subsequence of $r+1$ numbers represents a gap of length r.

The classical gap test considers the sequence $U_1, \ldots, U_n$ to be cyclic sequence with U_{n+j} identified with U_j. If m of the numbers $U_1, \ldots, U_n$ fall into the range $[\alpha, \beta)$, there are n gaps in the cyclic sequence. We have the following theorem:

Theorem B.2.3

Let t be some positive integer. Let G_r be the counts of the gaps of length r for $0 \leq r \leq t-1$, and G_t the count of gaps of length $\geq t$. Define $p = \beta - \alpha$, and let

$$p_r = p(1-p)^r, \text{ for } 0 \leq r \leq t-1 \tag{B.35}$$

$$pt = (1-p)^t. \tag{B.36}$$

then as $n \to \infty$, $V = \sum_{i=1}^{t} \frac{(G_i - np_i)^2}{np_i}$ follows a chi-squared distribution with t degrees of freedom.

The gap test is often applied with $\alpha = 0$ or $\beta = 1$ in order to omit one of the comparisons in the algorithm. The special cases $(\alpha, \beta) = (0, 1/2)$ or $(1/2, 1)$ give rise to tests that are sometimes called "runs above the mean" and "runs below the mean," respectively.

B.2.2.6 Maximum-of-t Test

For $0 \leq j < n$, let $V_j = \max(U_{tj}, U_{tj+1}, \ldots, U_{tj+t-1})$. Let $F(x)$ be the cdf of U_i, we now apply the Kolmogorov–Smirnov test to the sequence $V_0, V_1, \ldots, V_{n-1}$, with the distribution function $F_{max}(x) = F(x)^t, \ 0 \leq x \leq 1$.

To verify this test, we must show that the distribution function for the V_j is $F_{max}(x) = F(x)^t$. The probability that $\max(U_1, U_2, \ldots, U_t) \leq x$ is the probability that $\Pr(U_1 \leq x, U_2 \leq x, \ldots, U_t \leq x) = F(x)F(x)\cdots F(x) = F(x)^t$.

B.2.2.7 Runs Test

consider the sequence of ten digits "1298536704". Putting a vertical line at the left and right and between X_j and X_{j+1} whenever $X_j > X_{j+1}$, we obtain

$$|129|8|5|367|04|$$

This shows that run ups. Unlike the other tests, we should not apply a chi-squared test to the run counts, since adjacent runs are not independent. A long run will tend to be followed by a short run and vice versa. We can use other statistics to take this into account. However, a simple and more practical test is the following: If we "throw away" the element that immediately follows a run, so that when X_j is greater than X_{j+1} we start the next run with X_{j+2}, the run lengths are independent, and a simple chi-square test can be used.

Proposition B.2.1

For a sequence of iid continuous random variables, the probability of run of length k starting is given by

$$p_k = \frac{k}{(k+1)!} \tag{B.37}$$

For a run of length k, we need a $k+1$ random variables $X_1, \ldots, X_{k+1}$. If the sequence were iid, then we would have $(k+1)!$ possible orderings each with the same probability. The number of these orderings that give us a run of length k is exactly k. Since the number in the kth position must be the unique maximum of $X_1, \ldots, X_{k+1}$.

B.3 Additional Experimental Details

B.3.1 Additional Details

We evaluate the UMI function on the training data contained within https://olmo-data.org/dolma-v1_5r1/cc_en_head/cc_en_head-0000.json.gz.

Data	Avg. Size	Value
Generated by top-p	1000	0.0092
Generated by top-k	1000	0.0163
Different Temp. T	1000	0.0185
Random Tokens	2499	0.2617
Random Characters	1000	0.194

Table B.1: **Evaluating the value function for Llama2-7B** We report the values given by the UMI value function for different categories of data, alongside the average sequence sizes within each category.

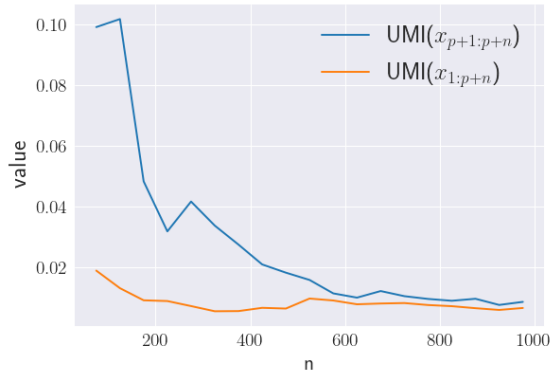

Figure B.1: UMI value function when the prompt is not given.

B.3.2 Temperature

Recall that the logits z_i produced by the language model (one for each token in our vocabulary) are transformed into probabilities through the softmax function given by:

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}, \tag{B.38}$$

The temperature T is a parameter used to adjust the output probabilities. A high temperature ($T > 1$) flattens the probability distribution bringing it closer to a uniform distribution. Meanwhile, a small T exacerbates any leads high probability tokens have making the probability distribution sharper. In other, temperature can be interpreted as the confidence a model has in its most likely responses. With temperature adjustment, the new probabilities

are given by

$$p_i = \frac{e^{z_i/T}}{\sum_{j=1}^{n} e^{z_j/T}}. \tag{B.39}$$

B.3.2.1 Sampling Methods

We give a brief description of the most common sampling methods used in text generation using language models.

Greedy Sampling: The next token is always chosen to be the highest probability token, as predicted by the model. While simple and fast, this sampling method is prone to producing sentences with low probability.This occurs because we might select a high-probability token initially, only to encounter subsequent tokens of considerably lower probabilities.

Beam Search: This method addresses the shortcomings of greedy sampling by maintaining a fixed number of potential sentences (sequences of tokens of some fixed length) called beams. It then chooses the sentence with the highest probability.

Top-k: This method restricts our selection of the next token to the top k tokens with the highest probability, where k is a hyper-parameter. Since k is fixed, this method does not dynamically adjust the number of candidates based on the probability distribution. As a result, some very unlikely tokens may be selected among those top k candidates.

Top-p: This method addresses the issues of top-k sampling by selecting as candidates the most probable tokens until their combined probability surpasses a specified threshold p.

As mentioned before, we can reduce each of the above sampling method to multinomial sampling methods with a modified probability distribution. For example, we can consider greedy sampling to be a multinomial sampling method with a distribution which places a probability of 1 on the token with the highest probability as predicted by the model.

B.3.3 On the choice of ϵ and α

Our choice for ϵ was guided by bounds on the statistical distance between the uniform distribution and its empirical distribution such as the one provided by the Dvoretzky–Kiefer–Wolfowitz inequality (see the remark on page 6) which gives $\|G_n - G\|_\infty \leq \sqrt{\frac{\ln(2/\delta)}{2n}}$, and (2) our own observations on the size of the statistical distance when we compute apply Rosenblatt's transformation on data on data generated by the model. A value of ϵ that is too small would make us miss those cases that are uniform but dependent. If we made ϵ too large, then we would flag many samples that do not have a marginal uniform distribution. The value of ω is one based on intuition that the value assigned to samples that are uniform and dependent should be larger than those that are uniform but independent ($\omega > \epsilon$), and less than those that are not uniform.

Appendix C

Omitted Details From Chapter 5

C.1 Proofs

C.1.1 Proof of Theorem 5.4.1

Let $\boldsymbol{Y} = \boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{C}) + \boldsymbol{W}$ be as in (5.11). $\{\widehat{\boldsymbol{Q}}(\boldsymbol{Y}) = \boldsymbol{Q}'\}$ is the event that our estimator outputs a different orientation $\boldsymbol{Q}'$ when the object is in orientation $\boldsymbol{Q}$. We can rewrite the average error in terms of events of this form.

$$\mathcal{L}(\boldsymbol{C}) = \sum_{\boldsymbol{Q} \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{W}} \left[\theta(\widehat{\boldsymbol{Q}}(\boldsymbol{Y}), \boldsymbol{Q}) \right], \tag{C.1}$$

$$= \sum_{\boldsymbol{Q} \in \mathcal{Q}} \sum_{\boldsymbol{Q}' \neq \boldsymbol{Q}} \mathbb{P}\left(\widehat{\boldsymbol{Q}}(\boldsymbol{Y}) = \boldsymbol{Q}' \right) \theta(\boldsymbol{Q}, \boldsymbol{Q}'). \tag{C.2}$$

The probability of the event $\{\widehat{\boldsymbol{Q}}(\boldsymbol{Y}) = \boldsymbol{Q}'\}$ is a difficult quantity to compute outside of cases when $\mathcal{Q}$ is very small. Therefore, we wish to upper bound it with another probability that is easier to compute. Recall that

$$\widehat{\boldsymbol{Q}}(\boldsymbol{Y}) = \arg\min_{\boldsymbol{Q} \in \mathcal{Q}} \|\boldsymbol{Y} - \boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{C})\|_F . \tag{C.3}$$

Given the functional form of the estimator, a necessary but insufficient condition for our estimator to output $\boldsymbol{Q}'$ is for the received signal $\boldsymbol{Y}$ to satisfy $\|\boldsymbol{Y} - \boldsymbol{F}(\boldsymbol{Q}; \boldsymbol{C})\| \geq \|\boldsymbol{Y} - \boldsymbol{F}(\boldsymbol{Q}'; \boldsymbol{C})\|$.

Given this relationship between the events, we have that:

$$\begin{aligned}&\{\widehat{\boldsymbol{Q}}(\boldsymbol{Y})\boldsymbol{Q}'\} \subseteq \\ &\{\, \|\boldsymbol{Y}-\boldsymbol{F}(\boldsymbol{Q};\boldsymbol{C})\|_F \geq \|\boldsymbol{Y}-\boldsymbol{F}(\boldsymbol{Q}';\boldsymbol{C})\|_F \,\}. \end{aligned} \tag{C.4}$$

The latter is event that our *noisy* channel output is closer to the *noiseless* channel output corresponding to $\boldsymbol{Q}'$ than the *noiseless* channel output corresponding to $\boldsymbol{Q}$. Combining (C.2) and (C.4) with the monotonicity of the probability function ($\mathbb{P}(A) \leq \mathbb{P}(B)$ for all events $A \subseteq B$), we obtain an upper bound on the average error.

$$\mathcal{L}(\boldsymbol{C}) = \sum_{\substack{\boldsymbol{Q}\in\mathcal{Q}\\ \boldsymbol{Q}'\neq\boldsymbol{Q}}} \mathbb{P}\left(\widehat{\boldsymbol{Q}}(\boldsymbol{Y})=\boldsymbol{Q}'\right)\theta(\boldsymbol{Q},\boldsymbol{Q}'), \tag{C.5}$$

$$\leq \sum_{\substack{\boldsymbol{Q}\in\mathcal{Q}\\ \boldsymbol{Q}'\neq\boldsymbol{Q}}} \mathbb{P}\big(\, \|\boldsymbol{Y}-\boldsymbol{F}(\boldsymbol{Q};\boldsymbol{C})\|_F \geq \|\boldsymbol{Y}-\boldsymbol{F}(\boldsymbol{Q}';\boldsymbol{C})\|_F \,\big) \\ \theta(\boldsymbol{Q},\boldsymbol{Q}'). \tag{C.6}$$

In other words, we use the standard pairwise error probability to develop an upper bound to the average error. Combining (C.6) with the following standard lemma [45], which provides the form for the probability of the event in the right-hand-side of (C.4), we obtain the desired result.

Lemma C.1.1

Let $\boldsymbol{Y} \sim \mathbb{C}\mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I}_n)$*, and let* $\boldsymbol{a} \in \mathbb{C}^n$*, then*

$$\mathbb{P}\left(\|\boldsymbol{Y}\| \geq \|\boldsymbol{Y}-\boldsymbol{a}\|\right) = \frac{1}{2}\mathrm{erfc}\left(\frac{\|\boldsymbol{a}\|}{2\sqrt{2}\sigma}\right), \tag{C.7}$$

where erfc *is the complement of the Gauss error function.*

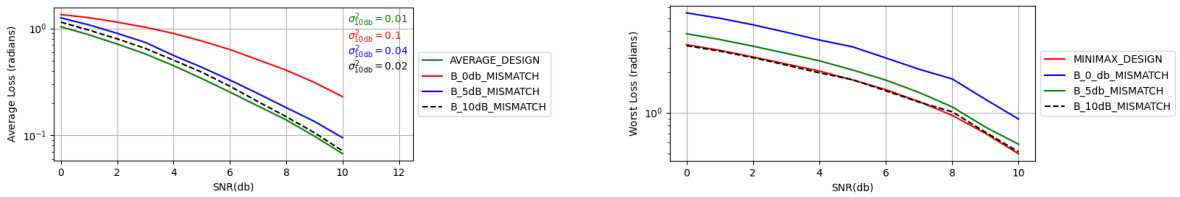

Figure C.1: The average (top) and worst-case (bottom) errors of the designed codes for different levels of channel estimation errors.

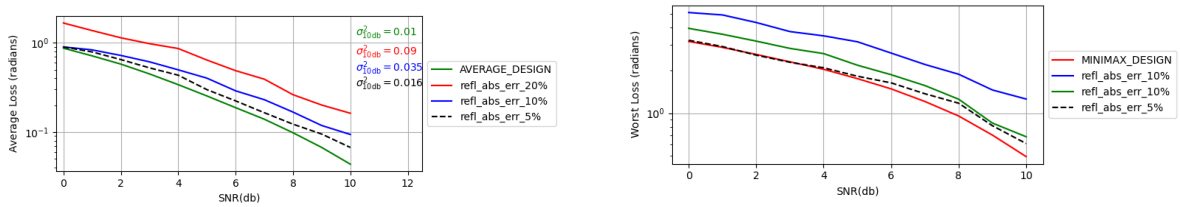

Figure C.2: The average (top) and worst-case (bottom) errors of the designed codes for different levels of channel estimation errors.

C.1.2 Proof of Theorem 5.4.2

Let $\boldsymbol{C} \in [m]^{N\times T}$ be a coding matrix, and let $\mathcal{P}$ be the family of distributions given by

$$\mathcal{P} = \{\mathbb{CN}(\boldsymbol{F}(\boldsymbol{Q};\boldsymbol{C}), \sigma^2\boldsymbol{I}); \boldsymbol{Q} \in \mathcal{Q}\}. \tag{C.8}$$

The probability distributions in $\mathcal{P}$ can be parameterized by their orientation. Let $P, P' \in \mathcal{P}$ be distributions with $\boldsymbol{Q}$ and $\boldsymbol{Q}'$ as their corresponding orientations, then Le Cam's method [62] asserts that:

$$\mathcal{M}(\boldsymbol{C}) \geq \frac{\theta(\boldsymbol{Q},\boldsymbol{Q}')}{2}\left(1 - \mathrm{TV}(P,P')\right), \tag{C.9}$$

where $\mathrm{TV}(P,P')$ is the total variation distance [14]. The Kullback-Leibler divergence [29] is a different and widely used measure of discrepancy between two probability distributions. It has many desirable properties, and it's an upper bound for many other popular discrepancy measures. Through Pinsker's inequality [14], one can upper bound the total-variation distance through the Kullback-Leibler divergence. The total-variation distance between two

probability distributions P and P', the Kullback-Leibler divergence, and Pinsker's inequality are respectively given by:

$$\mathrm{TV}(P, P') = \int_{\mathbb{C}^{KT}} |P(\boldsymbol{x}) - P'(\boldsymbol{x})| \, \mathrm{d}\boldsymbol{x}. \tag{C.10}$$

$$\mathrm{KL}(P||P') = \int_{\mathbb{C}^{KT}} P(\boldsymbol{x}) \log \frac{P(\boldsymbol{x})}{P'(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}. \tag{C.11}$$

$$\|P - P'\|_{\mathrm{TV}} \leq \sqrt{\frac{1}{2}\mathrm{KL}(P||P')}. \tag{C.12}$$

$$\begin{aligned}
\mathcal{M}(\boldsymbol{C}) &\geq \frac{\theta(\boldsymbol{Q}, \boldsymbol{Q}')}{2} \left(1 - \mathrm{TV}(P, P')\right), && \text{(C.13)} \\
&\geq \frac{\theta(\boldsymbol{Q}, \boldsymbol{Q}')}{2} \left(1 - \sqrt{\frac{1}{2}\mathrm{KL}(P||P')}\right), && \text{(C.14)} \\
&\geq \frac{\theta(\boldsymbol{Q}, \boldsymbol{Q}')}{4} \exp\left(-\mathrm{KL}(P||P')\right), && \text{(C.15)} \\
&= \frac{\theta(\boldsymbol{Q}, \boldsymbol{Q}')}{4} \exp\left(-\frac{\|\boldsymbol{F}(\boldsymbol{Q}; \boldsymbol{C}) - \boldsymbol{F}(\boldsymbol{Q}'; \boldsymbol{C})\|_F^2}{2\sigma^2}\right). && \text{(C.16)}
\end{aligned}$$

(C.14) follows from Pinsker's inequality, and (C.15) follows from the inequality $1 - x \leq e^{-x}$. For two independent Gaussians $P = \mathbb{C}\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$ and $P' = \mathbb{C}\mathcal{N}(\boldsymbol{\mu}', \sigma^2 I)$ with the same co-variance matrix, the Kullback-Leibler divergence is given by $\|\boldsymbol{\mu} - \boldsymbol{\mu}'\|^2 / 2\sigma^2$ [14], and therefore, we obtain (C.16). Since (C.16) holds for every $P, P' \in \mathcal{P}$, we can take the maximum over every pair of distributions (equivalently, every pair of orientations $\boldsymbol{Q}, \boldsymbol{Q}' \in \mathcal{Q}$), which gives

$$\mathcal{M}(\boldsymbol{C}) \geq \max_{\boldsymbol{Q}, \boldsymbol{Q}' \in \mathcal{Q}} \exp\left(\frac{-\|\boldsymbol{F}(\boldsymbol{Q}; \boldsymbol{C}) - \boldsymbol{F}(\boldsymbol{Q}'; \boldsymbol{C})\|_F^2}{2\sigma^2}\right) \frac{\theta(\boldsymbol{Q}, \boldsymbol{Q}')}{4}. \tag{C.17}$$

C.1.3 Proof of Theorem 5.4.3

Letting $\mathcal{P}$ be the same family of distributions as in (C.8), P and P' distributions in $\mathcal{P}$ with $\boldsymbol{Q}$ and $\boldsymbol{Q}'$ as their corresponding distributions, then we can start with Equation (C.14) which was obtained with Le Cam's method and Pinsker's inequality. In other words, we start with:

$$\mathcal{M}(\boldsymbol{C}) \geq \frac{\theta(\boldsymbol{Q}, \boldsymbol{Q}')}{2}\left(1 - \sqrt{\frac{1}{2}\mathrm{KL}(P||P')}\right), \tag{C.18}$$

where $\mathrm{KL}(P||P')$ is the Kullback-Leibler Divergence. We obtain (C.18) using Pinsker's inequality [14]. For two independent Gaussians $P = \mathbb{C}\mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$ and $P' = \mathbb{C}\mathcal{N}(\boldsymbol{\mu}', \sigma^2\boldsymbol{I})$ with the same covariance matrix, $\mathrm{KL}(P, P') = \|\boldsymbol{\mu} - \boldsymbol{\mu}'\|_2 / 2\sigma^2$. Hence,

$$\mathcal{M}(\boldsymbol{C}) \geq \frac{\theta(\boldsymbol{Q}, \boldsymbol{Q}')}{2}\left(1 - \frac{1}{2\sigma}\|\boldsymbol{F}(\boldsymbol{Q}; \boldsymbol{C}) - \boldsymbol{F}(\boldsymbol{Q}'; \boldsymbol{C})\|_F\right). \tag{C.19}$$

Now, let $\boldsymbol{R}_t$ be the diagonal matrix of reflectivities when using code configuration $\boldsymbol{c}_t$, then recall

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{Q}; \boldsymbol{c}_t) &= \boldsymbol{H}_{\boldsymbol{Q}}\boldsymbol{R}_t(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{R}_t)^{-1}\boldsymbol{H}_{\boldsymbol{Q}}^T\boldsymbol{s}_t, && \text{(C.20)}\\
&= \boldsymbol{H}_{\boldsymbol{Q}}(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{R}_t)^{-1}\boldsymbol{R}_t\boldsymbol{H}_{\boldsymbol{Q}}^T\boldsymbol{s}_t, && \text{(C.21)}\\
&= \boldsymbol{H}_{\boldsymbol{Q}}(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{R}_t)^{-1}\mathrm{diag}\left\{\boldsymbol{H}_{\boldsymbol{Q}}^T\boldsymbol{s}_t\right\} r(\boldsymbol{c}_t), && \text{(C.22)}
\end{aligned}$$

where $\boldsymbol{r}(\boldsymbol{c}_t)$ is the column vector of reflectivities corresponding to code $\boldsymbol{c}_t$. We wish to bound $\|\boldsymbol{f}(\boldsymbol{Q}; \boldsymbol{c}_t) - \boldsymbol{f}(\boldsymbol{Q}'; \boldsymbol{c}_t)\|$, so we use the inequality $\|\boldsymbol{A}_t\boldsymbol{x}_t\| \leq \|\boldsymbol{A}_t\|_F \|\boldsymbol{x}_t\|$ with $\boldsymbol{x}_t = \boldsymbol{r}(\boldsymbol{c}_t)$ and

$$\begin{aligned}
\boldsymbol{A}_t =& \boldsymbol{H}_{\boldsymbol{Q}}(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{R}_t)^{-1}\mathrm{diag}\{\boldsymbol{H}_{\boldsymbol{Q}}\boldsymbol{s}_t\}\\
&- \boldsymbol{H}_{\boldsymbol{Q}'}(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{R}_t)^{-1}\mathrm{diag}\{\boldsymbol{H}_{\boldsymbol{Q}'}\boldsymbol{s}_t\}.
\end{aligned}$$

Lemma C.1.2

If D is an approximate range between the tagged object's hinge point and the aperture, then

for all $t \in \{1, \ldots, T\}$

$$\|A_t\|_F \leq \left(\frac{K}{4\pi\lambda D^2}\right) \left\|\boldsymbol{X}^{\text{tag}}\right\|_F \left\|\widetilde{\boldsymbol{B}}_t\right\|_F \theta(\boldsymbol{Q}, \boldsymbol{Q}'). \tag{C.23}$$

Letting $\delta = \sqrt{\theta(\boldsymbol{Q}, \boldsymbol{Q}')}$, we have that:

$$\left\|\boldsymbol{F}(\boldsymbol{Q}; \boldsymbol{C}) - \boldsymbol{F}(\boldsymbol{Q}'; \boldsymbol{C})\right\|_F \tag{C.24}$$

$$= \sqrt{\sum_{t=1}^{T} \|\boldsymbol{f}(\boldsymbol{Q}; \boldsymbol{c}_t) - \boldsymbol{f}(\boldsymbol{Q}'; \boldsymbol{c}_t)\|} \tag{C.25}$$

$$\leq \sqrt{\sum_{t=1}^{T} \|\boldsymbol{A}_t\|_F^2 \|\boldsymbol{r}(\boldsymbol{c}_t)\|^2} \tag{C.26}$$

$$\leq \delta \sqrt{\sum_{t=1}^{T} \underbrace{\left(\frac{K}{2\pi\lambda D^2} \left\|\boldsymbol{X}^{\text{tag}}\right\|_F \left\|\widetilde{\boldsymbol{B}}_t\right\|_F \|\boldsymbol{r}(\boldsymbol{c}_t)\|\right)^2}_{\Delta_t}}, \tag{C.27}$$

where (C.26) follows from the inequality $\|\boldsymbol{A}\boldsymbol{x}\| \leq \|\boldsymbol{A}\|_F \|\boldsymbol{x}\|$, and (C.27) follows from Lemma C.1.2. Hence, a lower bound on the worst-case error is given by:

$$\mathcal{M}(\boldsymbol{C}) \geq \frac{\delta^2}{2} \left(1 - \frac{\delta}{2\sigma} \sqrt{\sum_{t=1}^{T} \Delta_t}\right). \tag{C.28}$$

To obtain a tightest bound, we have to find the maximum of the right-hand side with respect to δ. Maximizing (C.28) with respect to δ is equivalent to finding:

$$\max_x \quad x^2 (1 - x) \tag{C.29}$$

(C.29) attains it's maximum value at $x = 2/3$. This gives us that:

$$\mathcal{M}(\boldsymbol{C}) \geq \frac{32\pi^2\lambda^2\sigma^2 D^4}{27K^2 \left\|\boldsymbol{X}^{\text{tag}}\right\|_F^2 \sum_{t=1}^{T} \|\widetilde{\boldsymbol{B}}_t\|_F^2 \|\boldsymbol{r}(\boldsymbol{c}_t)\|_2^2} \tag{C.30}$$

C.1.4 Proof of Lemma C.1.2

For ease of computation, we drop the t subscript for now, and define $\widetilde{\boldsymbol{B}} = (\boldsymbol{I} - \boldsymbol{B}\boldsymbol{R}_t)^{-1}$ with $\tilde{b}_{i,j} = \tilde{\boldsymbol{B}}_{i,j}$, and $\tilde{\boldsymbol{H}}_{\boldsymbol{Q}} = \boldsymbol{H}_{\boldsymbol{Q}}\tilde{\boldsymbol{B}}$ with $\tilde{h}^{\boldsymbol{Q}}_{i,j} = (\tilde{\boldsymbol{H}}_{\boldsymbol{Q}})_{i,j}$. Taking $\mathbf{s}_t = \mathbf{1}$, we have that:

$$\|\boldsymbol{A}_t\|_F^2 = \sum_{k=1}^{K}\sum_{n=1}^{N}\left|\sum_{i=1}^{K}\tilde{h}^{\boldsymbol{Q}}_{k,n}h^{\boldsymbol{Q}}_{i,n} - \tilde{h}^{\boldsymbol{Q}'}_{k,n}h^{\boldsymbol{Q}'}_{i,n}\right|^2, \tag{C.31}$$

$$\leq \sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{i=1}^{K}\left|\tilde{h}^{\boldsymbol{Q}}_{k,n}h^{\boldsymbol{Q}}_{i,n} - \tilde{h}^{\boldsymbol{Q}'}_{k,n}h^{\boldsymbol{Q}'}_{i,n}\right|^2. \tag{C.32}$$

Using Cauchy-Schwartz inequality, we arrive at:

$$\left|\tilde{h}^{\boldsymbol{Q}_1}_{k,n}h^{\boldsymbol{Q}_1}_{i,n} - \tilde{h}^{\boldsymbol{Q}_2}_{k,n}h^{\boldsymbol{Q}_2}_{i,n}\right|^2 \leq \left\|\tilde{\mathbf{b}}_n\right\|_2^2\sum_{l=1}^{N}\left|h^{\boldsymbol{Q}_1}_{k,l}h^{\boldsymbol{Q}_1}_{i,n} - h^{\boldsymbol{Q}_2}_{k,l}h^{\boldsymbol{Q}_2}_{i,n}\right|^2 \tag{C.33}$$

Assuming we operate in the far-field region, we can approximate h with

$$h^{\boldsymbol{Q}}_{i,j} = \frac{e^{-j\left(\frac{2\pi}{\lambda}\right)\left\|\mathbf{x}_i^{ant} - \boldsymbol{Q}\mathbf{x}_j^{tag}\right\|_2}}{4\pi\left\|\mathbf{x}_i^{ant} - \boldsymbol{Q}\mathbf{x}_j^{tag}\right\|_2} \approx \frac{e^{-j\left(\frac{2\pi}{\lambda}\right)\left\|\mathbf{x}_i^{ant} - \boldsymbol{Q}\mathbf{x}_j^{tag}\right\|_2}}{4\pi D}, \tag{C.34}$$

where D is an approximate range between the tagged object's hinge point and the aperture. Hence, we can make the following approximation:

$$\left|h^{\boldsymbol{Q}}_{k,l}h^{\boldsymbol{Q}}_{i,n} - h^{\boldsymbol{Q}'}_{k,l}h^{\boldsymbol{Q}'}_{i,n}\right|^2 \approx \left(\frac{1}{4\pi D}\right)^4 \left|e^{-j\left(\frac{2\pi}{\lambda}\right)\left(\left\|\mathbf{x}_k^{ant} - \boldsymbol{Q}\mathbf{x}_l^{tag}\right\|_2 + \left\|\mathbf{x}_i^{ant} - \boldsymbol{Q}\mathbf{x}_n^{tag}\right\|_2\right)} - e^{-j\left(\frac{2\pi}{\lambda}\right)\left(\left\|\mathbf{x}_k^{ant} - \boldsymbol{Q}'\mathbf{x}_l^{tag}\right\|_2 + \left\|\mathbf{x}_i^{ant} - \boldsymbol{Q}'\mathbf{x}_n^{tag}\right\|_2\right)}\right|. \tag{C.35}$$

Using $1 - e^{-x} \leq x$, we can further upper bound (C.35) by:

$$\left(\frac{\sqrt{\frac{2\pi}{\lambda}}}{4\pi D}\right)^4 \left(\left\|\mathbf{x}_k^{ant} - \boldsymbol{Q}\mathbf{x}_l^{tag}\right\|_2 + \left\|\mathbf{x}_i^{ant} - \boldsymbol{Q}\mathbf{x}_n^{tag}\right\|_2 - \left\|\mathbf{x}_k^{ant} - \boldsymbol{Q}'\mathbf{x}_l^{tag}\right\|_2 - \left\|\mathbf{x}_i^{ant} - \boldsymbol{Q}'\mathbf{x}_n^{tag}\right\|_2\right). \quad \text{(C.36)}$$

Using the triangle and matrix norm inequalities in (C.36), we obtain

$$\left|h_{k,l}^{\boldsymbol{Q}} h_{i,n}^{\boldsymbol{Q}} - h_{k,l}^{\boldsymbol{Q}'} h_{i,n}^{\boldsymbol{Q}'}\right|^2 \leq 2\left(\frac{\sqrt{\frac{2\pi}{\lambda}}}{4\pi D}\right)^4 \left(\left\|\boldsymbol{x}_l^{\text{tag}}\right\|_2^2 + \left\|\boldsymbol{x}_n^{\text{tag}}\right\|_2^2\right) \theta(\boldsymbol{Q}, \boldsymbol{Q}'). \quad \text{(C.37)}$$

Combining (C.37) and (C.32), and summing over the indices of the tags and antennas, we obtain the desired result.

C.1.5 Proof of Lemma 5.4.1

Let $\boldsymbol{C} \in \mathbb{R}^{N \times T}$ be a code of size T, and let $\boldsymbol{\pi}$ be its corresponding proportions vector, then

$$\|\boldsymbol{F}(\boldsymbol{Q}; \boldsymbol{C}) - \boldsymbol{F}(\boldsymbol{Q}'; \boldsymbol{C})\|_2^2 \tag{C.38}$$

$$= \sum_{t=1}^{T} \|\boldsymbol{f}(\boldsymbol{Q}; \boldsymbol{c}_t) - \boldsymbol{f}(\boldsymbol{Q}'; \boldsymbol{c}_t)\|_2^2 , \tag{C.39}$$

$$= \sum_{t=1}^{T} \sum_{\boldsymbol{c} \in \{0,1\}^N} 1_{\boldsymbol{c}_t = \boldsymbol{c}} \|\boldsymbol{f}(\boldsymbol{Q}; \boldsymbol{c}) - \boldsymbol{f}(\boldsymbol{Q}'; \boldsymbol{c})\| , \tag{C.40}$$

$$= \sum_{\boldsymbol{c} \in \{0,1\}^N} \sum_{t=1}^{T} 1_{\boldsymbol{c}_t = \boldsymbol{c}} \|\boldsymbol{f}(\boldsymbol{Q}; \boldsymbol{c}) - \boldsymbol{f}(\boldsymbol{Q}'; \boldsymbol{c})\| , \tag{C.41}$$

$$= T \sum_{\boldsymbol{c} \in \{0,1\}^N} \frac{1}{T} \sum_{t=1}^{T} 1_{\boldsymbol{c}_t = \boldsymbol{c}} \|\boldsymbol{f}(\boldsymbol{Q}; \boldsymbol{c}) - \boldsymbol{f}(\boldsymbol{Q}'; \boldsymbol{c})\| , \tag{C.42}$$

$$= T \sum_{\boldsymbol{c} \in \{0,1\}^N} \boldsymbol{\pi}_{\boldsymbol{c}} \|\boldsymbol{f}(\boldsymbol{Q}; \boldsymbol{c}) - \boldsymbol{f}(\boldsymbol{Q}'; \boldsymbol{c})\| , \tag{C.43}$$

$$= T \langle \boldsymbol{\pi}, \boldsymbol{g}(\boldsymbol{Q}, \boldsymbol{Q}') \rangle . \tag{C.44}$$

Substituting (C.44) in the expressions of $\mathcal{U}(\boldsymbol{C})$ and $\mathcal{V}(\boldsymbol{C})$, we obtain the expressions in (5.20) and (5.21).

C.1.6 Proof of Theorem 5.4.4

Suppose $\boldsymbol{C} = [\boldsymbol{c}_1 \cdots \boldsymbol{c}_T]$ and $\boldsymbol{C}' = [\boldsymbol{c}'_1 \cdots \boldsymbol{c}'_T]$ are two coding matrices with the same relative weight vectors. In other words, $\boldsymbol{\pi}_{\boldsymbol{c}} = \boldsymbol{\pi}'_{\boldsymbol{c}}$ for every code configuration $\boldsymbol{c}$. If the counts of the different configurations are the same, then there exists some permutation τ on $\{1, \ldots, T\}$ such that $\boldsymbol{c}_{\tau(t)} = \boldsymbol{c}'_t$ for all t. We show the errors do not depend on the ordering of the code

configurations due to the iid nature of the noise. To see this, let $\boldsymbol{Y} = \boldsymbol{F}(\boldsymbol{Q}; \boldsymbol{C}) + \boldsymbol{W}$, then

$$\widehat{\boldsymbol{Q}}(\boldsymbol{Y}) \tag{C.45}$$

$$= \widehat{\boldsymbol{Q}}\left(\boldsymbol{F}(\boldsymbol{Q}; \boldsymbol{C}) + \boldsymbol{W}\right), \tag{C.46}$$

$$= \arg\min_{\boldsymbol{Q}'} \left\|\boldsymbol{F}(\boldsymbol{Q}; \boldsymbol{C}) - \boldsymbol{F}(\boldsymbol{Q}'; \boldsymbol{C}) + \boldsymbol{W}\right\|_F^2, \tag{C.47}$$

$$= \arg\min_{\boldsymbol{Q}'} \sum_{t=1}^{T} \left\|\boldsymbol{f}(\boldsymbol{Q}; \boldsymbol{c}_t) - \boldsymbol{f}(\boldsymbol{Q}'; \boldsymbol{c}_t) + \boldsymbol{w}_t\right\|^2, \tag{C.48}$$

$$= \arg\min_{\boldsymbol{Q}'} \sum_{t=1}^{T} \left\|\boldsymbol{f}(\boldsymbol{Q}; \boldsymbol{c}_{\tau(t)}) - \boldsymbol{f}(\boldsymbol{Q}'; \boldsymbol{c}_{\tau(t)}) + \boldsymbol{w}_{\tau(t)}\right\|^2, \tag{C.49}$$

$$= \arg\min_{\boldsymbol{Q}'} \sum_{t=1}^{T} \left\|\boldsymbol{f}(\boldsymbol{Q}; \boldsymbol{c}'_t) - \boldsymbol{f}(\boldsymbol{Q}'; \boldsymbol{c}'_t) + \boldsymbol{w}_{\tau(t)}\right\|^2, \tag{C.50}$$

$$= \widehat{\boldsymbol{Q}}\left(\boldsymbol{F}(\boldsymbol{Q}; \boldsymbol{C}') + \boldsymbol{W}'\right), \tag{C.51}$$

$$= \widehat{\boldsymbol{Q}}(\boldsymbol{Y}'), \tag{C.52}$$

where $\boldsymbol{w}_t$ is the t^{th} K-sized block of $\boldsymbol{W}$, $\boldsymbol{W}'$ is the vector with $\boldsymbol{w}'_t = \boldsymbol{w}_{\tau(t)}$, and $\boldsymbol{Y}' = \boldsymbol{F}(\boldsymbol{Q}; \boldsymbol{C}') + \boldsymbol{W}'$. Since the elements of $\boldsymbol{W}$ independent and identically distributed, we get that $\boldsymbol{W}$ and $\boldsymbol{W}'$ are also identically distributed. Combining the previous statement with the fact that $\boldsymbol{W}$ and $\boldsymbol{W}'$ are permutations of one another, we can replace any expectation with respect to $\boldsymbol{W}$ with an expectation with respect to $\boldsymbol{W}'$. Hence,

$$\mathcal{L}(\boldsymbol{C}) = \sum_{\boldsymbol{Q} \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{W}} \left[\theta(\widehat{\boldsymbol{Q}}(\boldsymbol{Y}), \boldsymbol{Q})\right], \tag{C.53}$$

$$= \sum_{\boldsymbol{Q} \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{W}} \left[\theta(\widehat{\boldsymbol{Q}}(\boldsymbol{Y}'), \boldsymbol{Q})\right], \tag{C.54}$$

$$= \sum_{\boldsymbol{Q} \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{W}'} \left[\theta(\widehat{\boldsymbol{Q}}(\boldsymbol{Y}'), \boldsymbol{Q})\right], \tag{C.55}$$

$$= \mathcal{L}(\boldsymbol{C}'). \tag{C.56}$$

(C.54) follows from the earlier computation, and (C.55) follows from the our earlier discussion. Repeating the exact same computation with $\sum_{\boldsymbol{Q} \in \mathcal{Q}}$ replaced with $\max_{\boldsymbol{Q} \in \mathcal{Q}}$ shows that

$\mathcal{M}(\boldsymbol{C}) = \mathcal{M}(\boldsymbol{C}')$ i.e. $\boldsymbol{C}$ and $\boldsymbol{C}'$ share the same worst-case performance. Hence, we get that $\boldsymbol{C}$ and $\boldsymbol{C}'$ share the same errors, and that our performance measures depend only on the coding matrix through the relative weight vector. It is worthwhile to note that this property of the errors is not unique to Gaussian noise, but is valid for any noise vector whose components are independent and identically distributed.

References

[1] Jayadev Acharya, Constantinos Daskalakis, and Gautam Kamath. Optimal testing for properties of distributions. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[2] Alessandro Achille, Giovanni Paolini, and Stefano Soatto. Where is the information in a deep neural network? *arXiv preprint arXiv:1905.12213*, 2019.

[3] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

[4] Ibrahim M Alabdulmohsin. Algorithmic stability and uniform generalization. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[5] Rubayet-E-Azim Anee and Nemai C. Karmakar. Chipless rfid tag localization. *IEEE Transactions on Microwave Theory and Techniques*, 61(11):4008–4017, 2013.

[6] Gregor Bachmann, Thomas Hofmann, and Aurelien Lucchi. Generalization through the lens of leave-one-out error. In *International Conference on Learning Representations*, 2022.

[7] Colby Boyer and Sumit Roy. — invited paper — backscatter communication and rfid: Coding, energy, and mimo analysis. *IEEE Transactions on Communications*, 62(3):770–785, 2014.

[8] Yuheng Bu, Shaofeng Zou, and Venugopal V. Veeravalli. Tightening mutual information based bounds on generalization error. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 587–591, 2019.

[9] Clément L. Canonne, Ayush Jain, Gautam Kamath, and Jerry Zheng Li. The price of tolerance in distribution testing. In *Proceedings of the 35th Annual Conference on Learning Theory (COLT 2022)*, COLT '22, July 2022.

[10] Kenneth Chang, Nathaniel Raymondi, Ashutosh Sabharwal, and Suhas N. Diggavi. Wireless paint: Code design for 3D orientation estimation with backscatter arrays. In *IEEE International Symposium on Information Theory, ISIT*, pages 1224–1229. IEEE, 2020.

[11] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

[12] Alvaro Collet, Dmitry Berenson, Siddhartha S. Srinivasa, and Dave Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *2009 IEEE International Conference on Robotics and Automation*, pages 48–55, 2009.

[13] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006.

[14] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

[15] Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression, 2024.

[16] Ilias Diakonikolas, Themis Gouleakis, John Peebles, and Eric Price. Sample-Optimal Identity Testing with High Probability. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41:1–41:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[17] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017.

[18] J.-L. Durrieu, J.-Ph. Thiran, and F. Kelly. Lower and upper bounds for approximation of the kullback-leibler divergence between gaussian mixture models. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4833–4836, 2012.

[19] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pages 642–669, 1956.

[20] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

[21] X. Fu, A. Pedross-Engel, D. Arnitz, and M. S. Reynolds. Simultaneous sensor localization via synthetic aperture radar (sar) imaging. In *2016 IEEE SENSORS*, pages 1–3, Oct 2016.

[22] Weihao Gao, Sewoong Oh, and Pramod Viswanath. Demystifying fixed k-nearest neighbor information estimators. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1267–1271, 2017.

[23] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2242–2251. PMLR, 09–15 Jun 2019.

[24] Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the science of language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[25] Michael M. Grynbaum and Ryan Mac. The times sues openai and microsoft over a.i. use of copyrighted work. *The New York Times*.

[26] Mahdi Haghifam, Jeffrey Negrea, Ashish Khisti, Daniel M Roy, and Gintare Karolina Dziugaite. Sharpened generalization bounds based on conditional mutual information and an application to noisy, iterative algorithms. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9925–9935. Curran Associates, Inc., 2020.

[27] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016.

[28] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1225–1234. JMLR.org, 2016.

[29] Peter Harremoes and Naftali Tishby. The information bottleneck revisited or how to choose a good distortion measure. In *2007 IEEE International Symposium on Information Theory*, pages 566–570, 2007.

[30] Hrayr Harutyunyan, Alessandro Achille, Giovanni Paolini, Orchid Majumder, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Estimating informativeness of samples with smooth unique information. In *International Conference on Learning Representations*, 2021.

[31] Hrayr Harutyunyan, Maxim Raginsky, Greg Ver Steeg, and Aram Galstyan. Information-theoretic generalization bounds for black-box learning algorithms. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24670–24682. Curran Associates, Inc., 2021.

[32] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

[33] GE Hinton and Drew van Camp. Keeping neural networks simple by minimising the description length of weights. 1993. In *Proceedings of COLT-93*, pages 5–13.

[34] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

[35] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. Towards efficient data valuation based on the shapley value. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1167–1176. PMLR, 16–18 Apr 2019.

[36] Chengkun Jiang, Yuan He, Xiaolong Zheng, and Yunhao Liu. Orientation-aware rfid tracking with centimeter-level accuracy. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 290–301. IEEE Press, 2018.

[37] Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C. Mozer. Characterizing structural regularities of labeled data in overparameterized models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5034–5044. PMLR, 2021.

[38] Juraj Juraska, Kevin Bowden, and Marilyn Walker. ViGGO: A video game corpus for data-to-text generation in open-domain conversation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 164–172, Tokyo, Japan, October–November 2019. Association for Computational Linguistics.

[39] Hoang Anh Just, Feiyang Kang, Tianhao Wang, Yi Zeng, Myeongseob Ko, Ming Jin, and Ruoxi Jia. LAVA: Data valuation without pre-specified learning algorithms. In *The Eleventh International Conference on Learning Representations*, 2023.

[40] Donald E. Knuth. *The art of computer programming, volume 2 (3rd ed.): seminumerical algorithms*. Addison-Wesley Longman Publishing Co., Inc., USA, 1997.

[41] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1885–1894. JMLR.org, 2017.

[42] Rasmus Krigslund, Petar Popovski, and Gert F. Pedersen. Orientation sensing using multiple passive rfid tags. *IEEE Antennas and Wireless Propagation Letters*, 11:176–179, 2012.

[43] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

[44] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Kumar Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee F Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Kamal Mohamed Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Joshua P Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah M Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham M. Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander T Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alex Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-LM: In search of the next generation of training sets for language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[45] Upamanyu Madhow. *Introduction to Communication Systems*. Cambridge University Press, USA, 1st edition, 2014.

[46] Pascal Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The Annals of Probability*, pages 1269–1283, 1990.

[47] Jeffrey Negrea, Mahdi Haghifam, Gintare Karolina Dziugaite, Ashish Khisti, and Daniel M Roy. Information-theoretic generalization bounds for sgld via data-dependent estimates. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[48] Gergely Neu, Gintare Karolina Dziugaite, Mahdi Haghifam, and Daniel M. Roy. Information-theoretic generalization bounds for stochastic gradient descent. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 3526–3545. PMLR, 15–19 Aug 2021.

[49] Maria-Elena Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454, 2006.

[50] Ki Nohyun, Hoyong Choi, and Hye Won Chung. Data valuation without training of a model. In *The Eleventh International Conference on Learning Representations*, 2023.

[51] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[52] Ankit Pensia, Varun Jog, and Po-Ling Loh. Generalization error bounds for noisy, iterative algorithms. *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 546–550, 2018.

[53] Yury Polyanskiy and Yihong Wu. Information theory: From coding to learning. 2024.

[54] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE, 2009.

[55] Murray Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952.

[56] Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1232–1240, Cadiz, Spain, 09–11 May 2016. PMLR.

[57] A. Sabharwal, P. Schniter, D. Guo, D.W̃. Bliss, S. Rangarajan, and R. Wichman. In-band full-duplex wireless: Challenges and opportunities. *IEEE Journal on selected areas in communications*, 32(9):1637–1652, 2014.

[58] Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. Learning 3-d object orientation from images. In *2009 IEEE International Conference on Robotics and Automation*, pages 794–800, 2009.

[59] Martin Scherhäufl, Markus Pichler, Erwin Schimbäck, Dominikus J. Müller, Andreas Ziroff, and Andreas Stelzer. Indoor localization of passive uhf rfid tags based on phase-of-arrival evaluation. *IEEE Transactions on Microwave Theory and Techniques*, 61(12):4724–4729, 2013.

[60] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[61] Ali Asghar Nazari Shirehjini, Abdulsalam Yassine, and Shervin Shirmohammadi. An rfid-based position and orientation measurement system for mobile objects in intelligent environments. *IEEE Transactions on Instrumentation and Measurement*, 61(6):1664–1675, 2012.

[62] A. N. Shiryayev and Yu. A. Koshevnik. Review: Lucien le cam, asymptotic methods in statistical decision theory. *Bull. Amer. Math. Soc. (N.S.)*, 20(2):280–285, 04 1989.

[63] Yu. M. Shtarkov. Asymptotic minimax regret for data compression, gambling, and prediction. *Problems of Information Transmission*, 23(3):3–17, 1987.

[64] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[65] Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American Journal of Mathematical and Management Sciences*, 23(3-4):301–321, 2003.

[66] Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[67] Elahe Soltanaghaei, Adwait Dongare, Akarsh Prabhakara, Swarun Kumar, Anthony Rowe, and Kamin Whitehouse. Tagfi: Locating ultra-low power wifi tags using unmodified wifi infrastructure. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(1), March 2021.

[68] Elahe Soltanaghaei, Akarsh Prabhakara, Artur Balanuta, Matthew Anderson, Jan M. Rabaey, Swarun Kumar, and Anthony Rowe. Millimetro: Mmwave retro-reflective tags for accurate, long range localization. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, MobiCom '21, page 69–82, New York, NY, USA, 2021. Association for Computing Machinery.

[69] Thad Starner, Bastian Leibe, David Minnen, Tracy Westyn, Amy Hurst, and Justin Weeks. The perceptive workbench: Computer-vision-based gesture tracking, object tracking, and 3d reconstruction for augmented desks. *Machine Vision and Applications*, 14:59–71, 01 2003.

[70] Thomas Steinke and Lydia Zakynthinou. Reasoning About Generalization via Conditional Mutual Information. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 3437–3452. PMLR, 09–12 Jul 2020.

[71] Stewart J. Thomas, Eric Wheeler, Jochen Teizer, and Matthew S. Reynolds. Quadrature amplitude modulated backscatter in passive and semipassive uhf rfid systems. *IEEE Transactions on Microwave Theory and Techniques*, 60(4):1175–1182, 2012.

[72] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan

Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[73] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, USA, 2005.

[74] Chandra Shekhara Kaushik Valmeekam, Krishna Narayanan, Dileep Kalathil, Jean-Francois Chamberland, and Srinivas Shakkottai. Llmzip: Lossless text compression using large language models, 2023.

[75] Georgios Vougioukas and Aggelos Bletsas. Doa estimation of a hidden rf source exploiting simple backscatter radio tags. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4355–4359, 2021.

[76] Georgios Vougioukas and Aggelos Bletsas. Doa estimation with a single antenna and a few low-cost backscattering tags. *IEEE Transactions on Communications*, 70(10):6849–6860, 2022.

[77] Teng Wei and Xinyu Zhang. Gyro in the air: tracking 3d orientation of batteryless internet-of-things. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 55–68. ACM, 2016.

[78] Geoffrey Wolfer and Aryeh Kontorovich. Minimax testing of identity to a reference ergodic markov chain. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 191–201. PMLR, 26–28 Aug 2020.

[79] Geoffrey Wolfer and Aryeh Kontorovich. Chapter 3 - learning and identity testing of markov chains. In *Artificial Intelligence*, volume 49 of *Handbook of Statistics*, pages 85–102. Elsevier, 2023.

[80] Zhaoxuan Wu, Yao Shu, and Bryan Kian Hsiang Low. DAVINZ: Data valuation using deep neural networks at initialization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24150–24176. PMLR, 17–23 Jul 2022.

[81] Qun Xie and A.R. Barron. Asymptotic minimax regret for data compression, gambling, and prediction. *IEEE Transactions on Information Theory*, 46(2):431–445, 2000.

[82] Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[83] Xinyi Xu, Zhaoxuan Wu, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Validation free and replication robust volume-based data valuation. In *Advances in Neural Information Processing Systems*, volume 34, pages 10837–10848. Curran Associates, Inc., 2021.

[84] Jiaxi Yang, Wenglong Deng, Benlin Liu, Yangsibo Huang, James Zou, and Xiaoxiao Li. Gmvaluator: Similarity-based data valuation for generative models, 2024.

[85] Po Yang, Wenyan Wu, Mansour Moniri, and Claude C. Chibelushi. Efficient object localization using sparsely distributed passive rfid tags. *IEEE Transactions on Industrial Electronics*, 60(12):5914–5924, 2013.

[86] Ya-xiang Yuan. A review of trust region algorithms for optimization. *ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics*, 09 1999.

[87] Weiping Zhu, Jiannong Cao, Yi Xu, Lei Yang, and Junjun Kong. Fault-tolerant rfid reader localization based on passive rfid tags. In *2012 Proceedings IEEE INFOCOM*, pages 2183–2191, 2012.