Identifying Feature, Parameter, and Sample Subsets in Machine Learning and Image Analysis

by

Ronak Mehta

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences Department)

at the

UNIVERSITY OF WISCONSIN–MADISON

2023

Date of final oral examination: 2023-03-10

The dissertation is approved by the following members of the Oral Committee:

Yong Jae Lee, Associate Professor, Computer Sciences
Frederic Sala, Assistant Professor, Computer Sciences
Michael Newton, Professor, Statistics
Vikas Singh (Advisor), Professor, Biostatistics and Medical Informatics

Acknowledgments

I was lucky enough to work with a number of amazing labmates, collaborators, and friends. In no particular order, the following people have contributed immensely via collaborations, discussions, and support in countless ways. Hyunwoo Kim, Seong Jae Hwang, Zirui Tao, Tianyi Shan, Haoliang Sun, Jurijs Nazarovs, Anita Sinha, Jeffery Kline, Sourav Pal, Vishnu Lokhande, Zhichun (Eric) Huang, Jeffery Kline, Sathya Ravi, Yunyang Xiong, Rudrasis Chakraborty. Won Hwa Kim, Vamsi Ithapu, Takis Chytas, Xingjian Chen, Zihang Meng were great labmates, and though we did not collaborate directly contributed significantly to making the lab a good place to work.

Michael Newton and Glenn Fung were immensely helpful as faculty mentors in providing both guidance and direct collaboration, and were it not for them this process may have been significantly more painful. I'm also thankful for the advice on specific projects given by Vivek Prabhakaran, Ming Yuan, and Sterling Johnson, as well as my committee for providing helpful feedback as this thesis developed.

The following friends, peers, and roommates contributed significantly to the atmosphere, and experience, and going home to and out with them at the end of a week or after a deadline was crucial to being able to do it again for the next. Michael O'Neill, Christopher Magnano, Ross Kleiman, Bryce Sandlund, Hollis Howe.

Therapy and medication aside, this would not have been possible without the continuous and unwavering support of my advisor Vikas. There were many times where I was uncertain, struggling, or concerned, and advice and discussions with him helped alleviate those feelings. Aside from the explicit and direct collaborations, either through whiteboard discussions or publication writing, those higher-level, personal conversations were crucial in carrying me over the finish line.

And lastly this work could not have been completed without a number of grants supporting myself and my collaborators. Work in Chapter 3 was supported in part by NIH grants R01 AG040396, AG021155, EB022883 and NSF grants DMS 1265202 and

CAREER award 1252725. The authors were also supported by the UW Center for Predictive Computational Phenotyping (via BD2K award AI117924) and the Wisconsin Alzheimer's Disease Research Center (AG033514). I was supported by a fellowship via training grant award T32LM012413. Work in Chapter 4 was supported by grants NSF CAREER award RI 1252725, UW CPCP (U54AI117924), R01AG059312, R01EB022883, RF1AG062336, and a NIH predoctoral fellowship to RM via T32 LM012413. Work in Chapter 5 was supported by grants from the National Institutes of Health numbered RF1AG059312, RF1AG062336 and RF1AG059869, NSF award CCF 1918211, funds from the American Family Insurance Data Science Institute at UW-Madison, and UIC-ICR start-up funds. Work in Chapter 6 was supported in part by NIH grants RF1AG059312, RF1AG062336, RF1AG059869, and NSF award CCF 1918211. The work presented here was partially developed while myself and collaborators Jeffery Kline and Glenn Fung were at the Machine Learning Group at American Family Insurance.

Contents

List of Figures

List of Tables

Abstract

Modern machine learning has proven to be extremely effective in aiding and automating a large number of tasks, beginning with simple image recognition, now ranging widely from full language translation and understanding to computer-aided medical diagnosis and drug discovery. These advances have largely been enabled by significant development in computation schemes and algorithms, that have come alongside exponential increases in the scale of training data available. Success in these cases is measured directly by performance, evaluating some sort of error or accuracy that proves to be competitive with even domain-level experts. Recent research within the field has thus now moved to orthogonal, but equally important questions involving robustness to data distribution shifts, model fairness, interpretability of model outputs, and explainability of model predictions. The analysis of these varied questions typically look at the learning formulation with a finer toothed comb, identifying individual elements or groups of elements of interest. These ideas all fall under a similar fundamental problem of *finding subsets*, where they be of groups within the population, features of the input, or sections of the model. Uniquely shaped by their particular machine learning instantiations, we need methods for identifying (**1**) subsets of training samples or subpopulations which have disparate outcomes for a given model, (**2**) feature subsets that are sufficient or uniquely explain a particular model prediction, and (**3**) important parameter subsets or paths through neural network models that explicitly describe how a model output was generated. In this dissertation, we will describe a number of methods for addressing these subset selection problems. Developing mathematical tools based primarily on aspects of differential geometry and conditional independence, we will demonstrate theoretical and empirical effectiveness of these methods on a wide breadth of problems that can be distilled in the manner above, including hypothesis testing with medical imaging, predicting disease progression, machine unlearning, and increasing model fairness.

Chapter 1

Introduction

Modern applications of machine learning in a broad range of industrial and consumer-facing systems have become ubiquitous. Most interactions with daily technologies now intrinsically involve a request to some “smart“ system in the “cloud”, where those interactions range from a request for map directions to simply loading a webpage. Neural network models, and the recent advances of deep learning, have enabled these systems that make such applications possible. These models have achieved human-level performance on learning tasks including image classification (He et al., 2016; Krizhevsky et al., 2017), image segmentation (Minaee et al., 2021), video analysis (Zhang et al., 2016), text understanding and generation (Devlin et al., 2018; Brown et al., 2020), and have slowly begun to solve more fundamental scientific problems such as protein folding (Noé et al., 2020) drug discovery (Vamathevan et al., 2019), and medical diagnosis (Bakator and Radosav, 2018). While this performance is largely attributed to model size, the abundance of high quality training data has equally contributed to real world performance, enabling model training over millions of real world samples (Russakovsky et al., 2015; Schuhmann et al., 2021), and potentially billions of synthetic samples via environment simulation (Silver et al., 2016).

While deployment in some domains (recommender systems, object detection) may benefit almost unconditionally from this vastly expanded capability, rightful hesitancy has limited their widespread use in particular applications where impacts on individuals, people, or environments may be at stake. These “last mile” concerns take a few forms. In mission critical applications such as medical diagnosis, the impact of an error can be extremely large, even if a misprediction happens extremely infrequently. Additionally, large scale model training and architecture search can require exorbitant amounts of energy producing high emissions, and their scale can limit participation to only large actors with vast existing resources (Anthony et al., 2020). The accessibility and effectiveness of these models can also vary significantly based on the training data, and disparate outcomes can be exacerbated by existing social inequity.

While existing human or "natural" systems that these models aim to assist are not perfect, our real world has developed norms and regulations that enable them to function. A medical diagnosis might require a physician to explain what symptoms led them to that particular conclusion. Energy metering and carbon taxes may be applied to limit emissions. Regulatory satisfaction may require analysis proving equal opportunity, or that specific protected classes are not used in decision making. These ideas are difficult to directly translate to automated machine learning systems, but proxies have been identified that we can build upon.

These norms and regulations answer a number of questions we may also try to pose to our machine learning models. What is the cost to learn this task? What led to this particular outcome? Why is this outcome different from another?

If the answer to these questions is negative or unknown, follow-up questions all take an interesting form: Can we learn a smaller model with similar performance? Can we identify the most important features? Which individuals or groups are being treated unfairly, and can we change that? These questions ask us to identify a *subset* of some relevant set, dependent on setting, and this identification is our focus here.

Taking a step back, let's take a look at a representative system. Figure 1.1 illustrates a typical learning pipeline. A dataset is collected and used to train a model, by minimizing the error over those samples in the dataset (top). A "sample" can be a single measured value, or it can be a large, highly structured object with many "features." The model is made up of some "parameters" that are tuned during training to learn a good predictor over the training dataset. This model is then used to predict, or *infer*, on new data seen "in the wild" (bottom). Our questions above are formally asking to identify *subsets of these objects*: is a subset of the model parameters sufficient for learning? Which subset of the features are important for a prediction? Which subset of the dataset exhibit a specific attribute?

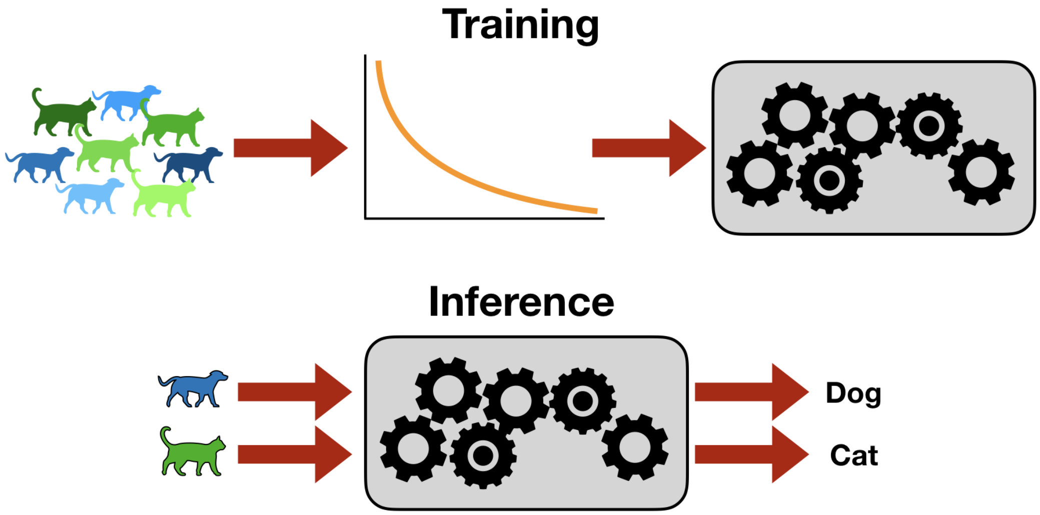


Figure 1.1: Machine learning training and inference visualization.

> ***This thesis** focuses its main efforts on identifying these important subsets of model, feature, and sample space, to enable answering questions necessary for mainstream adoption of machine learning methods.*

Let us step a bit deeper into a basic illustrative example. In order to ease understanding, we can first begin with a basic formulation of learning methods, from which the questions above can take specific forms. Learning methods typically try to identify a function mapping (model) that is able to complete a specific task at some high level of proficiency. Say we have some dataset comprising of sample pairs (x, y), where we wish to predict y from x. Our prediction, say $\hat{y}$, might be the output of some unknown function f that we attempt to learn from training data. Let our approximation to this function be $\hat{f}$. This can take many forms, based on assumptions and prior information we may have on the relationships among the data. Consider the simple *linear* case, where we want to learn some parameter w such that $y = w \cdot x$. Given n sample pairs (x_i, y_i) indexed by i, traditional statistics and optimization literature yields the following *least squares* problem formulation, where we want to minimize the "squared error" between the observed values y_i and the predicted $\hat{y}_i := w \cdot x_i$:

$$\hat{f} := \hat{w} = \arg\min_{w} \sum_{i=1}^{n} (y_i - w \cdot x_i)^2 \tag{1.1}$$

This formulation extends without much change to a multi-dimensional form of the input x and respectively, w: the canonical case where a number of features, say d, of x, or *covariates* (e.g., symptoms), are used together to predict the outcome (e.g., diagnosis). This is typically represented as a d-dimensional vector, with each position representing a different feature. If we are interested in which features of x are important, we can look at the relative values of the learned "weights" w. In this simple setting, the importance of a feature (say x^j) can be exactly determined by the importance of the corresponding parameter (w^j). A weight value far from zero may indicate that corresponding feature is important for diagnosis.

In this case and others, traditional statistical learning methods have been studied for many decades. Linear regressors, decision trees, and support vector machines have all been analyzed under this lens. New research focuses particularly on the differences associated with moving from classical *under-parameterized* models to modern (deep) **over-parameterized** models: where the model size vastly outnumbers the number of input samples. Methods for estimating the number of samples needed, the time to learn a particular task, and the generalization ability all require new perspectives in this regime. While nascent, these high-dimensional approaches attempt to fill the gap between statistical and deep models to enable similar measures of sample influence, feature importance, and model understanding.

A full picture. Let us expand our notation from the example above to consider this more general framing. Consider a dataset $X := \{x_i\}_{i=1}^n$ of size n where each data point x_i in the set X is drawn from some underlying distribution over the domain $x_i \sim \mathcal{X}^d$, with domain dimensionality (number of features) d (typically indexed by j as x^j). A model f is fit using a parameterization $\theta \in \Theta$, with Θ the space of possible parameterizations (models) with some intrinsic dimension p. Generalizing the least squares "error measure" from (1.1) to an arbitrary *loss* ℓ, we have

$$\hat{f} := \hat{\theta} = \underset{\theta \in \Theta}{\arg\min} \sum_{x_i \in X} \ell(f_\theta(x_i)) \tag{1.2}$$

From an analysis perspective, we might be interested in any one of (**a**) subsets of input features $\mathcal{F} \subseteq \mathcal{X}$ that are important for the downstream task, (**b**) associating model subsets $\mathcal{P} \subseteq \Theta$ with specific inputs or groups of inputs, or (**c**) subsets or subgroups of samples $S \subseteq X$ that are sufficient or representative of the entire dataset.

Crucially, an uninformed search for a subset is computationally infeasible. For

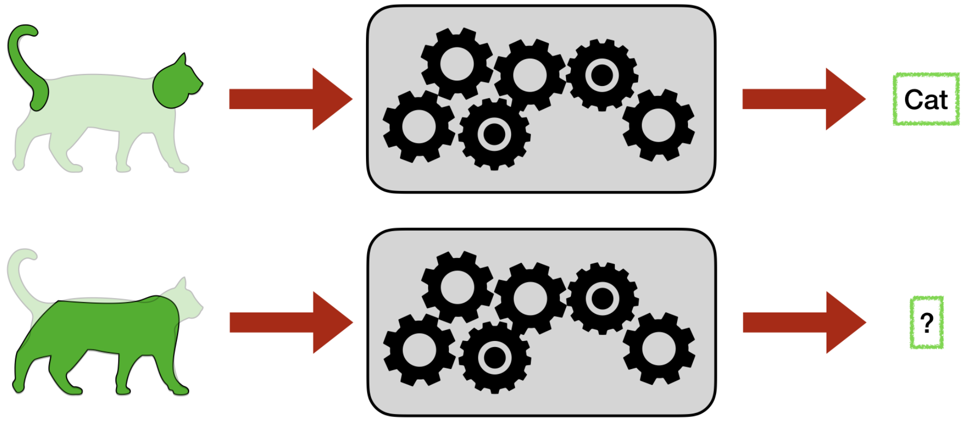


Figure 1.2: An example of identifying specific features important to the learning task.

a superset of size $n = |X|$, the set of all subsets is the power set, with a size of 2^n! If an identification procedure requires looking over all of these and choosing a "best" one by some measure, the procedure will be limited to very small supersets. Efficient methods have been developed in each of the three contexts above to avoid this exponential search.

Feature Selection. From the statistics side, penalized weighting of parameters in models similar to Equation (1.1) has been analyzed thoroughly, proving effective *sparse* recoveries, with nonzero elements identifying the selected features. Correlation and other statistical measures such as mutual information can be used to identify features that relate most to the outcome of interest based on the chosen measure. Analyses of variance and covariance have been widely used to determine if statistically significant relationships among variables of interest. When the number of variables is large, but an assumption can be made about their internal structure, scan statistics (Glaz et al., 2001; Chan and Walther, 2013) allow for a structured "scanning" over the input space, skipping subsets of features unlikely to demonstrate relationships based on the measure of interest. Adaptations of sensitivity analysis, via noise addition and perturbations have found success (Yeung et al., 2010; Zhang and Wallace, 2015), alongside activation mapping (Zhou et al., 2016; Selvaraju et al., 2017). These methods typically generate an analagous "weighting" over the input space, identifying features most salient for the specific task (Figure 1.2).

These weighting approaches work well with more complex models f compared to

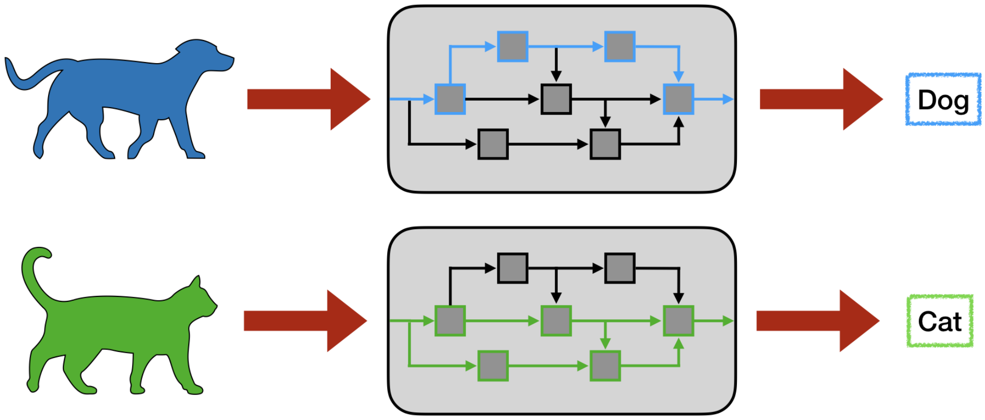


Figure 1.3: An example of identifying specific parameters important to the learning task.

the linear case above: newer "black-box" methods have been developed for identifying important features. Recent popular methods with broad applicability include SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-agnostic Explanations) (Lundberg and Lee, 2017; Ribeiro et al., 2016). Shapley methods estimate the importance, or weight, of a particular feature for a prediction through randomized masking to estimate marginal contributions. LIME samples the model space around the input of interest to estimate the decision function locally, penalizing complexity of this estimation to enable interpretability of the explanations. Predominantly created and deployed in computer vision and imaging domains, these methods have proven effective in identifying parts of the input that drive the function output. In models that generalize well, these features correspond to regions of the image that a human would use in performing the same task.

Parameter Selection. Selection in the model space generally takes two forms. First, as a prior, restriction, or assumption over the model space, and second, as a post-hoc method for an "explainable" proxy. Regularization, sparsity, and gating methods are often used independent of the type or size of the model, to encourage the solution to fall within a specific region of the model space. A complete picture of the theoretical underpinnings of these methods in deep learning has not yet been identified, but some progress has nonetheless explained their effectiveness in practice (Hardt et al., 2016; Jacot et al., 2018; Neyshabur et al., 2014). Outside of the actual training process, a number of methods have been proposed for *model* selection. By far the most valuable approach in scalable machine learning has been architecture search, enabling a higher-level identification of *which* parameters should be learned, and what their relationship should be (Elsken et al., 2019). With a model learned, identifying a simpler model takes the form of *knowledge distillation*. Originally presented in Hinton et al. (2015) and overviewed for modern DNNs in Gou et al. (2021), the technique involves training a simpler "student" model to mimic the output of the original, complex "teacher" model. To aid interpretability, *disentanglement* approaches have been widely successful, with the goal of associated specific, distinct, parameters or regions of model space with salient factors of variation within the data (Creager et al., 2019; Locatello et al., 2019). Text and image generation models have benefited largely from similar ideas, enabling tunable "knobs" corresponding to independent features of interest (Higgins et al., 2017; Karras et al., 2019; Hjelm et al., 2019).

Of particular interest here are the parameters relevant to specific regions of the input space *after* training (Figure 1.3). Here, recent analysis of deployed networks has shown that indeed, models tend to learn subsets of parameters corresponding to specific subregions of the input space, or subtasks (Bau et al., 2017a; Fong and Vedaldi, 2018), and current work continues to explore these network regions to aid in interpretability and explainability (Bau et al., 2017b; Zhou et al., 2018), some associating specific paths through the network with specific tasks (Geiger et al., 2022; Elhage et al., 2021).

Sample Selection. Interest in identification of a specific sample within the training set, or a sample at test time, have led to a number of other approaches. Identification in the most classical sense takes the form of outlier detection. Traditional statistics would use some form of the largest error to the model fit, perhaps after removing that sample and refitting. In the deep learning setting, we do not have the luxury of being able to train a new model for each sample we wish to evaluate. Thus methods for black-box outlier detection have been developed (Huang et al., 2020; Ren et al., 2019). Outliers within the data and embedding space notwithstanding, the *influence* of a particular sample on learning is of independent interest: are some samples particularly valuable or detrimental to training? Is there a minimal set sufficient for learning the task? Building on ideas in archetypal analysis (Cutler and Breiman, 1994) and coreset identification (Agarwal et al., 2005; Ravi et al., 2019), post-training

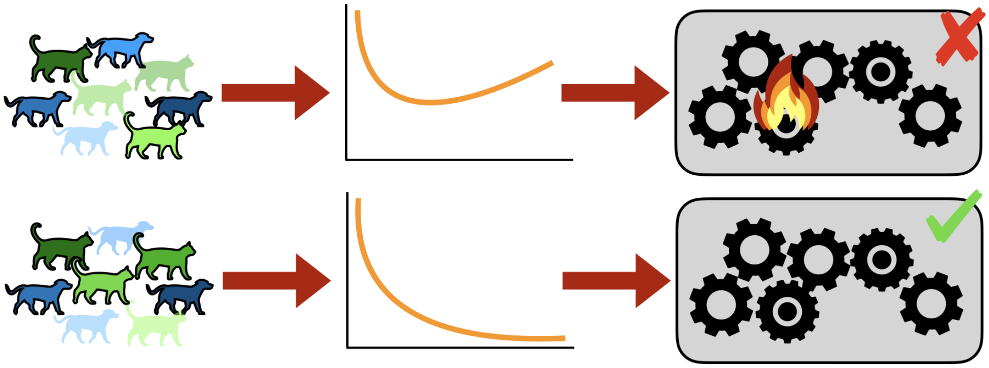


Figure 1.4: An example of identifying specific samples important to the learning task.

methods have also been effective in understanding sample influence (Koh and Liang, 2017; Golatkar et al., 2020a; Huang et al., 2020).

Additionally, once a sample or set of samples has been identified, it may be the case that we wish to adjust our model to either increase or decrease their influence or some other measure. In the context of fairness, a subset of samples may be "outliers", but represent a true mode of the sample space that has not been fit properly. Then we may want "in-the-loop" methods that can adjust for this individual or group unfairness ("outlierness") during training (Mehrabi et al., 2021). After a model has been trained however, post-hoc adjustments are difficult. One topical application of this is machine *unlearning*, where we wish to remove a sample's influence completely without retraining (Bourtoule et al., 2021; Cao and Yang, 2015). As we will see, this is still yet impractical with existing methods, and we will use novel tools to construct practical algorithms for efficient post-hoc model updates based on a sample's influence.

Thesis Goal: *Identify, construct, and evaluate methods for* ***efficient*** *subset identification in modern machine learning feature, model, and input spaces.*

1.1 A Few Motivating Examples

Consider a traditional machine learning classification task in which we would like to predict whether an individual has a specific disease condition based on a medical resonance image (MRI) scan of their brain. Our input feature x may consist of a 3D-array of values in $\mathbb{R}^{\mathcal{I}\times\mathcal{J}\times\mathcal{K}}$ measuring some intensity of the imaging modality at each voxel, indexed by a tuple $(i, j, k) \in (\mathcal{I}, \mathcal{J}, \mathcal{K})$. Our outcome variable y may simply be a binary label of whether the input scan has been labeled by a radiologist as one demonstrating typical disease characteristics. Using an off the shelf 3D convolutional neural network with adjustments to match our input size, we can very quickly set up and train a system to predict disease presence with a high degree of accuracy.

Example 1. With a prediction for a specific scan, or predictions over a number of scans, we might be interested in identifying which regions of the brain are most important for diagnosis. These regions, $R \subset \mathcal{R} := \mathbb{R}^{\mathcal{I}\times\mathcal{J}\times\mathcal{K}}$, can be specific groups of pixels in the image that may correspond to known functional networks. Methods such as attention and class activation maps may work here, but there are a few issues. The number of samples available to learn a model is very small compared to the both the dimension of the input and the number of parameters in the model, i.e., $n \ll p$ and $n \ll d$. Thus it is very easy to overfit, and for areas of interest to be associated with intricacies of particular input data rather than true, real differences defined by the disease.

Furthermore, recent medical imaging studies have moved past simple difference detection: trends over time, and the ability to predict *future* disease development have by far become the setting of most interest. Given an image of a healthy individual, is it possible to predict what their scan, or their future disease diagnosis, may be up to 10, 20, or more years in the future? If a number of scans have been collected over some timeframe, can the *trajectory* of the individuals' development be extrapolated to estimate progression? As traditional models extended for temporal analysis grow in both size and complexity, a number of subproblems explicitly related to model and input subspaces arise. In this thesis we address two such problems: **statistically rigorous identification of temporally evolving subsets**, and **characterizations of deep models that enable efficient training of recurrent models with large scale time-varying data**.

Example 2. With the rapid growth of AI and machine learning applications has come valid concerns regarding both guarantees of privacy. Recent technology legislation has made the importance clear in all aspects of data use, and particular projects and groups have demonstrated that machine learning is not independent of this need (Harvey, 2021). A new issue raised within this intersection is the "right to be forgotten". If a model has been trained with a particular users' data, they should have some recourse or right to both remove their data from the training set, and also know that the model has not learned from their data. On the surface, this poses a significant problem for model builders and organizations that spend large amounts of time and resources in training deep learning models.

In the medical imaging example above this is especially important: with fewer samples it is more likely that information about any particular one could "leak", and the model's performance may degrade significantly as a relatively large percentage of it's training data is removed. Thus tailored methods must be developed to ensure both privacy and performance, without requiring full retraining. As we will see, **identification of model parameter subsets** that are particularly important for a particular sample's influence in a model enables *efficient machine unlearning*.

Example 3. From an alternative perspective, we may want to identify specific samples rather than have them specified a priori. Traditionally a rigorous area of study under classical statistics, outlier detection and accounting have become a subfocus for many within the machine learning community as well (Golatkar et al., 2020a,b; Huang et al., 2020; Ren et al., 2019). While subgroups of input samples may be outliers, it is more often the case that they represent known heterogeneity within the data. These differences may be marked using group information known a priori, and most learning tasks aim to learn tasks in a *subgroup-independent* manner. In our disease prediction model above, these groups could simply be stratified by the type of scanner used to acquire the image, but it could also be a systematic difference correlated with some protected or delineating attribute. Commonly used atlases for brain MRI registration are often constructed from the scans of predominantly right-handed middle-aged adults (Fonov et al., 2009). Resulting downstream analysis may be significantly worse for people outside that age range or those who are left-handed. These types of biases can directly lead to disparate performance and results on *all* individuals outside of that group. Optimization and regularization methods with this focus come under the umbrellas of model fairness, enforcing invariance, and spurious feature regularization among others. However, many existing methods do not scale well to larger models or as the number of subgroups grows, as is often the case when intersections of protected classes must be considered. Here we identify and construct a particular solution for **groupwise fairness that enables efficient in-the-loop fairness regularization**.

> *Here we focus our effort on identifying these important subsets of model, feature, and sample space for feature association, model size reduction, model unlearning, and, fairness. Specifically, taking advantage of both existing statistical and geometric methods, we develop new methods for localizing subsets in a range of settings from hypothesis testing to deep learning.*

1.2 Thesis Scope and Contributions

We explore the intersections of classical statistical and geometric constructions with modern machine learning methods. Figure 1.5 shows the overall scope projected along three axes: feature, parameter, and sample spaces. Below we briefly introduce the main problems studied in this thesis.

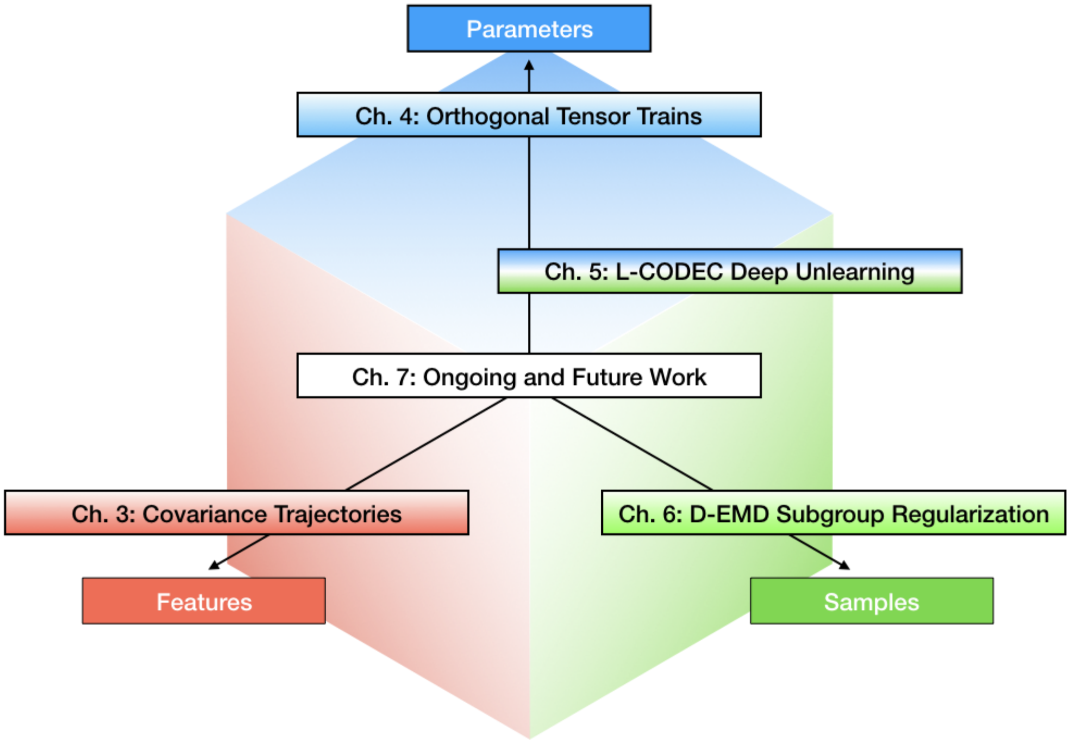

Parameters
Ch. 4: Orthogonal Tensor Trains
Ch. 5: L-CODEC Deep Unlearning
Ch. 7: Ongoing and Future Work
Ch. 3: Covariance Trajectories
Ch. 6: D-EMD Subgroup Regularization
Features
Samples

Figure 1.5: Thesis scope, projected over three representative axes.

Chapter 3: Second-Order Modeling and Group Difference Analysis over Time

Results in coupled or temporal graphical models offer schemes for estimating the relationship structure between features when the data come from related (but distinct) longitudinal sources (Zhou et al., 2010; Qiu et al., 2016). A novel application of these ideas is for analyzing group-level differences, i.e., in identifying if *trends* of estimated objects (e.g., covariance or precision matrices) are different across disparate conditions (e.g., gender or disease). Often, poor effect sizes make detecting the *differential* signal over the *full* set of features difficult: dependencies between only a *subset of features* may manifest differently across groups. We first suggest a parametric model for estimating trends in the space of SPD matrices as a function of one or more covariates. With this in hand, we move on to using graphical model approaches to determine on which covariates sets it is most meaningful to fit these trends. We generalize scan statistics to graph structures, and search over distinct **subsets of features** (graph partitions) whose temporal dependency structure may show statistically significant group-wise differences. We theoretically analyze the Family Wise Error Rate (FWER) and bounds on Type 1 and Type 2 error. On a cohort of individuals with risk factors for Alzheimer's disease (but otherwise cognitively healthy), we find scientifically interesting group differences and associated subsets, where the default analysis, i.e., models estimated on the full set of features, do not survive reasonable significance thresholds.

Chapter 4: Efficient Tensor Representations for Feasible Temporal Deep Learning

Modern deep networks have proven to be very effective for analyzing real world images. However, their application in medical imaging has required additional and specific machinery, primarily due to the large dimension of three-dimensional images, requiring enormous network architectures – if we treat an image (and not image patches) as a sample. These issues only compound when the focus moves towards longitudinal analysis through recurrent structures, and when a point estimate of model parameters is insufficient in scientific applications where a reliability measure is necessary. Building upon the differential geometry insights from the previous chapter, we will adapt a particular tensor decomposition, the tensor train, to construct networks with significantly fewer parameters. Using the theoretical guarantees afforded by its geometry, it enables us to train powerful recurrent networks on whole brain image volumes. Resembling **model and parameter selection** methods such as compression and distillation, we analyze the *orthogonal tensor train*, and demonstrate its ability to represent a standard network layer both theoretically and empirically. We demonstrate its ability to effectively reconstruct whole brain volumes with faster convergence and stronger confidence intervals compared to the standard tensor train decomposition. We provide code and show experiments on the ADNI dataset using image sequences to regress on a cognition related outcome.

Chapter 5: Practical Unlearning via Large-Scale Conditional Independence Testing

Recent legislation has led to interest in *machine unlearning*, i.e., removing specific training samples from a *predictive* model as if they never existed in the training dataset. Unlearning may also be required due to corrupted/adversarial data or simply a user's updated privacy requirement. For models which require no training (k-NN), simply deleting the closest original sample can be effective. But this idea is inapplicable to models which learn richer representations. Recent ideas leveraging optimization-based updates scale poorly with the model dimension d, due to inverting the Hessian of the loss function. We describe a variant of a new conditional independence coefficient, L-CODEC, to identify a **subset of the model parameters** with the most semantic overlap on an individual sample level. Our approach completely avoids the need to invert a (possibly) huge matrix. By utilizing a Markov blanket selection, we find that L-CODEC is also suitable for deep unlearning, as well as other applications in vision. Compared to alternatives, L-CODEC makes approximate unlearning possible in settings that would otherwise be infeasible, including vision models used for face recognition, person re-identification and NLP models that may require unlearning samples identified for exclusion.

Chapter 6: Reducing Subgroup Fairness via High Dimensional Earth Mover's Distances

Optimal transport has recently emerged as a useful tool for machine learning through its connections with geometry, statistical machine learning, and through practical algorithms. Existing methods that leverage optimal transport often regularize using a Wasserstein metric or by computing barycenters, for example. We leverage optimal transport, except that we take advantage of a recently-introduced algorithm that computes a generalized earth mover's distance. Not only is this algorithm computationally cheaper to compute compared to existing barycentric measures, but our method has the additional advantage that gradients used for backpropagation can be directly read off of the forward pass computation, which leads to substantially faster model training. This speedup enables practical methods of accounting for large **subgroups of samples** that may need to be treated equally when stratified. We provide technical details about this new regularization term and its properties, and experimental demonstrations of improved training speed over existing Wasserstein-style methods.

Chapter 7: Conclusions and Future Work

The thesis will conclude with brief descriptions of both completed and ongoing work that has been strongly motivated by the methods developed herein. Chapters 3 and 4 above have been extended in a number of directions, and continues to influence future research direction. The more recent work in Chapters 5 and 6 is serving as an ongoing foundation for new work. Developments include developing and providing a plug-and-play efficient optimal transport tool for community use, addressing a important gap in existing off-the-shelf methods. Particularly motivated by the general shift towards ensuring models such as GPT-2 and Stable Diffusion are fair and equitable, we conclude with a brief reflection on the role of conditional independence as a measure of influence and its role in interpretability.

1.3 Outline

Chapter 2 covers the essential background necessary for the developments presented in the following chapters, including specifics of graphs and hypothesis testing, as well as relevant modern methods for learning and optimization. In Chapters 3 through 7, we describe four perspectives to address subset identification. Chapter 3 explores and focuses on the identification of feature subsets varying over time. In Chapter 4 we describe a method of constraining the parameter space in a particular manner that enables more efficient large scale neural networks. Next, Chapter 5 provides a solution to the machine unlearning problem, enabled through a particular conditional independence parameter selection scheme, vastly reducing network update costs. Chapter 6 ends with a unique solution to subgroup fairness, where we take advantage of an efficient solution to the d-dimensional earth mover's problem to regularize large models when the number of subgroups can be large. We conclude in Chapter 7 with ongoing and future work, as well as some final thoughts.

Chapter 2

Background

Here we will briefly describe some background concepts and ideas that will aid and facilitate discussion of subset identification in later chapters. Following some general notation, we will begin with an overview of classical probability and statistics, and a focus on the hypothesis testing schemes built upon in the third chapter. This is followed by the basics of Riemannian differential geometry and the geometry of tensor objects, the key objects that enable the tests and efficiencies described in both Chapters 3 and 4. Next we will provide an overview of optimization tools and methods particularly suited and developed for neural network methods, alongside typical forms and objectives in the machine learning literature, used in Chapter 5. We conclude this chapter with a discussion on optimal transport methods in deep learning, important for our discussion of novel optimization schemes in Chapter 6.

2.1 General Notations

The following notations will be standard throughout the sequel, with any particular chapter overloads or redefinitions explicitly mentioned.

- Calligraphic capital letters such as $\mathcal{D}, \mathcal{M}, \mathcal{X}$ will typically represent spaces.
- Lowercase letters such as i, x, n will represent vectors as well as scalars (indices, sizes, and dimensions), and subsequently a vector $v \in \mathbb{R}^d$ would be a d-dimensional vector over the reals. The transpose will be indicated by $\top$.
- Capital letters such as X, H, T will refer to matrices and higher order tensors, as well as sets (including random variables) based on context.

- We use $\mathbb{R}^d$ to represent d-dimensional vector space over the reals. The positive orthant of $\mathbb{R}^d$ is denoted $\mathbb{R}^d_+ := \left\{x \in \mathbb{R}^d : x(i) \geq 0, i \in [d]\right\}$. $e := (1, \ldots, 1) \in \mathbb{R}^d$ will denote the constant vector.

- For learning settings, the training dataset will typically be defined by S, with samples $x_i \in S$ in the general setting, and $(x_i, y_i) \in S$ in the supervised setting. Other datasets T, U may also be used and defined in context.

- Subscripts will denote indexing into sets or vectors, or estimation (in the case of integrals and expectations) over the specified subscript. Superscripts will be used as additional decorators when needed.

Additional theoretical and experimental details can typically found in the Appendix, with specific references in appropriate sections of the main text.

2.2 Probability and Independence

A density function (distribution) is a mapping from each element in an outcome space $x \in X$ to a number between 0 and 1. This mapping is generally written as $p(x) := \mathbb{P}(X = x), x \in X$ to denote the distribution over possible realizations x of X. X may also denote a *set* of random variables $X := \{X_i\}_{i=1}^n$, where a draw from the distribution over X results in a vector x. When X follows the law defined by a specific $p(x)$, we say X is a random variable with distribution $p(x)$.

For $q \geq 1$, we define the q-norm as $\|x\|_q := \left(\sum_{i \in [n]} |x(i)|^q\right)^{1/q}$, and if q is suppressed, then $\|x\| := \|x\|_2$. A discrete probability distribution is a point $p \in \mathbb{R}^n_+$ with $e'p = \|p\|_1 = 1$.

The *expectation* of a random variable $\mathbb{E}[X]$ is the "weighted average" over all outcomes. For discrete variables (which will be our main focus with expectations),

$$\mathbb{E}[X] = \sum_{x \in X} x \cdot p(x) \tag{2.1}$$

The properties of distributions below extend to measures of expectations among multiple variables, with some caveats mentioned in specific chapters as needed.

Over a set of random variables X and Y, the distribution $p(x, y) := \mathbb{P}(X = x, Y = y)$ is the *joint* distribution over both variables (X, Y). The *marginal* distribution for x

is computed by "summing out" y, i.e.,

$$p(x) = \mathbb{P}(X = x) = \sum_y \mathbb{P}(X = x, Y = y) = \sum_y p(x, y) \tag{2.2}$$

$$= \int p(x, y)dy \tag{2.3}$$

We say that the random variables X and Y are *independent* if their joint distribution is equal to the product of their marginals: $p(x, y) = p(x)p(y)$. Intuitively, a draw from one distribution has no impact on the draw from the other. In the case where the variables are *dependent*, the distributions are linked. The dependent draw is defined by the *conditional* distribution $p(y|x)$. Then, a joint distribution can factor as an iterative draw first from $p(x)$ followed by from $p(y|x)$: $p(x, y) = p(x)p(y|x)$. Importantly, this dependence is commutative: it is also true that $p(x, y) = p(y)p(x|y)$. When $p(y|x) = p(y)$, we say that Y is independent of X. This commutativity leads to the formulation of Bayes' Rule:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \tag{2.4}$$

This form is used throughout statistics and machine learning. In most settings, we are interested in estimating some parameters θ, or distribution over parameters $p(\theta)$, given a distribution over data $p(x)$. Applying Bayes' Rule:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} \tag{2.5}$$

Where $p(x|\theta)$ is the *likelihood* of observing the data x given some parameter set, $p(\theta)$ is the *prior* assumption on the base distribution over parameters, $p(x)$ is the marginal over all data (typically treated as a normalizing constant), and $p(\theta|x)$ is the *posterior* distribution over the parameters given some data. Alternatively, we wish to update our estimation of the parameters θ from $p(\theta)$ to $p(\theta|x)$ given some observations x that change our beliefs about the parameter space. Maximum a posteriori (MAP) estimation attempts to find the parameters that maximize this posterior, using Bayes' rule for tractable computation and estimation.

$$\max_\theta p(\theta|x) = \max_\theta \frac{p(x|\theta)p(\theta)}{p(x)} = \max_\theta p(x|\theta)p(\theta) \tag{2.6}$$

If we have some samples $\{x_i\}_{i=1}^n$ from the data distribution that comprise a dataset, then we have the following equivalent log-transformed model:

$$\max_\theta \sum_{i=1}^{n} \log p(x_i|\theta) + \log p(\theta) \tag{2.7}$$

This is the canonical form used in most of machine learning: generally associating a "loss" or recovery term to the likelihood, and a regularization to the prior.

Conditional Independence

With three or more variables, independence relations among variables may be determined by which variables are being conditioned upon.

Definition 2.1 (Conditional Independence)**.** *For three random variables X, Y, Z, we say that Y is conditionally independent of X given Z, written as $Y \perp\!\!\!\perp X|Z$, if*

$$p(y, x|z) = p(y|z)p(x|z). \tag{2.8}$$

Intuitively, once Z is known, X provides no additional information in predicting Y. This can also be written as $p(y|x, z) = p(y|z)$.

Graphs. With a larger number of variables, tracking independence relations can become cumbersome with this notation. *Graphs* are commonly used in place. Graphs G are defined by their vertex and edge sets $G := (E, V)$. The vertices V correspond to some random variables $X, Y, Z, \ldots$, or $X_1, \ldots, X_n$. An *undirected* graph is one in which the existence of an edge e_{ij} implies the existence of edge e_{ji}. Within a *probabilistic graphical model*, an edge e_{ij} implies a *conditional dependence.* The omission of a particular edge thus has an explicit meaning:

$$e_{ij} \notin E \iff X_i \perp\!\!\!\perp X_j|X_{V\setminus(i,j)} \tag{2.9}$$

Where $V \setminus (i, j)$ is the rest of the variables in the set and graph. A graph with these independence relations is also referred to as a Markov graph: pairwise conditional independence relations imply *global* conditional independence relations: for any sets of variables A, B, C, if C "separates" A and B, While directed graphs (Bayesian networks) are used and are of independent interest, in this thesis we focus on undirected graphs.

Multivariate Gaussians. Extending the typical univariate Gaussian distribution $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$ defined by mean and variance parameters μ, σ^2, we have the following exponential density function for multivariate Gaussian distributions over n variables, with realizations as the vector $\mathbf{x} := [x_1, \ldots, x_n]^\top$:

$$p(x_1, \ldots, x_n) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left(\frac{1}{2}(\mathbf{x} - \mu)^\top \Sigma^{-1} (\mathbf{x} - \mu)\right) \tag{2.10}$$

Where μ is the vector of means for each individual variable, and Σ is the $n \times n$ *covariance* matrix describing the second-order interactions among the variables. The tractability of both computing the density and estimating it in typical likelihood estimation frameworks makes the multivariate Gaussian a particular attractive prior used in many applications, including estimating independence. Specifically,

Theorem 2.2. *Let X be governed by a multivariate Gaussian as defined in* (2.10). *Then it holds that*

$$\Sigma_{ij} = 0 \iff X_i \perp\!\!\!\perp X_j \tag{2.11}$$

Complete independence among variables is often not possible and in most settings not valuable. However, a more practical result for conditional independence exists.

Theorem 2.3. (*Lauritzen*, *1996*) *Let X be governed by a multivariate Gaussian as defined in* (2.10), *and $\Omega := \Sigma^{-1}$ be the precision matrix. Then it holds that*

$$\Omega_{ij} = 0 \iff X_i \perp\!\!\!\perp X_j | X_{\setminus(i,j)} \tag{2.12}$$

This immediately yields that the precision matrix encodes the edges of the probabilistic graphical model over the variables. Estimating these dependencies is thus equivalent to estimation of the precision matrix. Further, if the precision matrix is sparse, we can derive dependencies between features when the data is high-dimensional and/or the number of measurements are small.

Estimating Parameters and Measures Over and Among Distributions

Multivariate data analysis exploiting the conditional independence structure between features or covariates using undirected graphical models is now standard within any data analysis toolbox. When samples are drawn from a particular family of distributions, a number of methods exist to estimate the parameters that fit that data best. The estimation of a graphical model has been extensively studied and a rich literature is available describing its statistical and algorithmic properties (Koller and Friedman, 2009; Jordan, 1998). MAP estimation and maximum likelihood estimation as described above are typically used, but in many cases their standard forms do not yield parameters that reveal independence, i.e., it is often impossible for standard methods to result in a parameter estimate of zero.

To determine the relationship among two random variables, measures such as the Pearson correlation coefficient, and Spearman and Kendall rank coefficients (Myers et al., 2013) are frequently used, with values close to zero suggesting a low importance, and a value close to 1 suggesting perfect dependence. However, these measures are only applicable with specific assumptions about the possible dependencies between the variables: Pearson coefficients can only identify linear dependencies, and rank coefficients typically fail with non-monotonic dependencies (e.g., periodic functions). These methods can be computed pairwise among all variables among a set of variables with size greater than 2, however, to estimate conditional dependencies typically requires the estimation of *partial* coefficient measures. Additionally, while results similar to Theorem 2.3 exist under specific assumptions , practical estimation does not often yield exact zeros, and estimating all partial coefficients separately can be computationally heavy.

With some assumptions, *sparse* recoveries are possible while estimating ALL conditional independencies. Following the results from multivariate Gaussians above, a *penalized* version of a maximum likelihood estimate can be recovered through the following *graphical lasso* (Friedman et al., 2008) formulation:

$$\hat{\Omega} = \min_{\Omega \succeq 0} \mathrm{tr}(\hat{\Sigma}\Omega) + \log|\Omega| + \lambda||\Omega||_1 \tag{2.13}$$

With $\hat{\Sigma}$ being the sample covariance of the data samples, and λ a penalization weight. The constraint $\Omega \succeq 0$ restricts the solution to be positive semi-definite: symmetric matrices with nonnegative eigenvalues. This ℓ_1 penalization has been shown to be

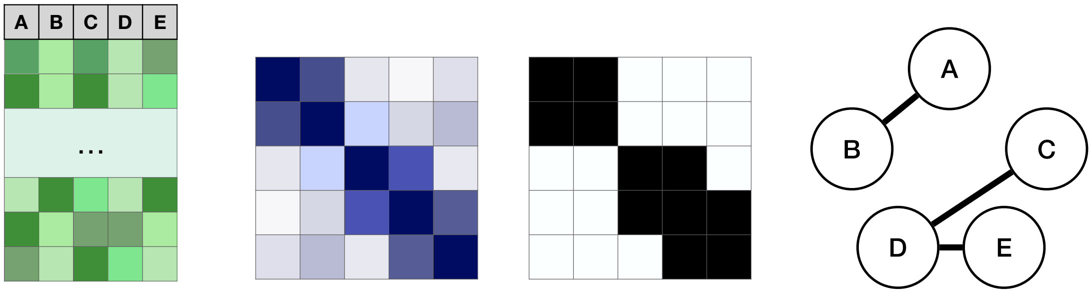


Figure 2.1: From data collected on various measures (left), we can compute a covariance matrix, estimate a sparse inverse, and recover the conditional independence relationships among measures (right).

easy to compute, and a number of alternative versions have been proposed with varying theoretical properties and recovery guarantees (Cai et al., 2011; Yuan, 2010). Particularly interesting are extensions to *nonparanormal* distributions, which allow for a much larger set of graphs to be estimated over other distributions via a rank covariance matrix, generalizing the pairwise rank and distance coefficients above (Liu et al., 2009; Xue and Zou, 2012).

The distributional assumptions needed for all of these methods, however, are often completely unknown in practice, or the data represent highly complex densities and functions that cannot be represented by simple exponential families or rank-based measures. Newer measures of *nonparametric* estimations of dependence have been developed, such as distance correlation and kernel-based coefficients (Székely and Rizzo, 2014; Wang et al., 2015; Doran et al., 2014), but they tend to either only test for independence and not strength of functional relationship, or have complex instantations or asymptotic theories, making them difficult to deploy and rely upon in practice.

A recent coefficient, the Chatterjee coefficient, has been demonstrated to have a number of desirable properties, and has been further developed into an elegant method for testing conditional independence (Chatterjee, 2020; Azadkia and Chatterjee, 2019). Particularly, no conditional densities need to be estimated, it can be computed in $O(n \log n)$ time where n is the number of data samples, it asymptotically to 0 for conditional independence, and 1 for measurable functions, and it requires *absolutely no assumptions* on the law over the random variables. For arbitrary variables random X, Y, Z, where Y is univariate and X, Z can be multivariate of any size, the conditional coefficient is given by

$$T(Y, X|Z) = \frac{\int \mathbb{E}\left[\mathrm{VAR}(\mathbb{P}(Y \geq t)|X, Z)|Z)\right] d\mu(t)}{\int \mathbb{E}\left[\mathrm{VAR}(\mathbb{I}\{Y \geq t\}|Z\right] d\mu(t)} \tag{2.14}$$

with $\mathrm{VAR}(\cdot)$ denoting the variance over the random variable. In Chapter 5 we will take advantage of these properties. Critically, without a specific procedure, identifying the conditioning set for a particular independence test is exponential. If we wish to find which variables $X_1, \ldots, X_n$ are sufficient for creating independence between some outcome Y and the rest of the variables, naïvely we would need to test all possible subsets. An advantage of the coefficient of dependence in Azadkia and Chatterjee (2019) is that it admits a linear time algorithm for iteratively building the sufficient set, naturally enabling an algorithm for constructing a Markov graph over the variables of interest.

Hypothesis Testing

Statistical hypothesis testing involves formally defining and testing a hypothesis about the world. A prior "null" assumption is defined. The *null* hypothesis, H_0 represents the default assumption or expectation that there is no relationship or distinction among the true population states, whereas the *alternative* hypothesis, H_A, describes the world in which some hypothesized relationship or distinction does exist. Testing proceeds by collecting observations of the variables of interest, computing a *test statistic*, and comparing that statistic against a prior *null distribution*. If the test statistic is larger than a predefined critical value, the null hypothesis is rejected: there is reasonable evidence to suggest the alternative may be true. When we fail to reject the null, there is insufficient evidence to support the alternative claim.

The form of the test statistic and null distribution are defined by the specific hypothesis being tested, as well as the prior assumptions about the parameters of interest and data collected.

Example: Testing a difference of means. Say we have collected samples from two different groups, representing the height of each person in the group. Our task is to determine if the average height of the two groups is significantly different from each other. Naïvely, we can compute the averages of the two groups and compare. In practice these averages will never be equal, so how can we more rigorously determine if we should consider some measured difference significant enough to say the groups are different? We can set up the following hypothesis test.

Let μ_1 be the true *population mean* of group 1, and μ_2 the true population mean of group 2. Our hypotheses are:

$$H_0 : \mu_1 = \mu_2 \qquad H_A : \mu_1 \neq \mu_2 \tag{2.15}$$

Let us assume we have collected the same number of samples from each group, n, and that the means of the collected samples are $\hat{\mu}_1$ and $\hat{\mu}_2$, and the standard deviations are s_1 and s_2. Then the test statistic

$$t = \frac{\hat{\mu}_1 - \hat{\mu}_2}{\sqrt{\frac{s_1^2 + s_2^2}{n}}} \tag{2.16}$$

follows a t-distribution with $n-1$ degrees of freedom, **if** there is no difference between the means. Comparing the value of the measured statistic over the observed data to the corresponding t-distribution allows us to determine how likely or unlikely it is that our observation follows the law that the means are equal. If we want our test to accurately identify a difference between means 95% of the time, we can set our threshold t^* (critical value) for rejecting the null to be the point where $\mathbb{P}(t \geq t^*) \geq 0.95$.

Importantly, the value of the t-statistic and corresponding distribution and critical value are extremely dependent on the number of samples n acquired for the test. As we will see, in cases where the number of samples is very small and our hypothesis describes a subtle difference between groups, novel tests and procedures are necessary to effectively identify those differences.

Likelihood ratios and permutation testing

Two challenges often appear in practice: **(1)** An obvious test statistic may not exist, and **(2)** the null distribution of that statistic or the one chosen may not have a clear form.

In the case of large parametric models, distributions over hundreds of parameters or more become infeasible to compute in practice. In these cases, the *likelihood-ratio test* can be used. With our posterior notation above, we have the following test statistic for the null hypothesis $\theta \in \Theta_0, \Theta_0 \subseteq \Theta$ and the alternative $\theta \in \Theta^C := \Theta \setminus \Theta_0$:

$$\lambda_{LR} = -2\ln \left[\frac{\sup_{\theta \in \Theta_0} p(x|\theta)}{\sup_{\theta \in \Theta} p(x|\theta)} \right] \tag{2.17}$$

The true power of the LRT statistic comes from two fundamental results. First, the distribution of λ_{LR} asymptotically approaches a χ^2 distribution with a fixed number of degrees of freedom, enabling easy testing (Wilk's Theorem, (Wilks, 1938)). And second, the Neyman-Pearson lemma states that this likelihood-ratio test is the most powerful α level test for this case (Neyman et al., 1933).

When even a likelihood ratio cannot be constructed in this form, a nonparametric test exists that may be used for any statistic defined by the problem of interest. Consider again a test of the form in (3.3), but where instead of the mean we have some arbitrary statistic over our two groups. A *permutation test* comprises of generating a null distribution through resampling. By shuffling the data and recomputing the statistic, we can estimate the distribution of the statistic if there were no difference among the groups. This distribution can then be used as the null distribution for checking if the true group allocations result in a test statistic significantly different from the null distribution.

These ideas will be used to construct and evaluate hypothesis tests in Chapter 3, as well as in subsequent work building upon those results.

2.3 Differential Geometry

Here we present a brief overview of differential geometry. For a more in-depth background, we refer interested readers to Do Carmo (1992), Lee (2003), and Spivak (1981).

Differentiable manifold. A *differentiable* (*smooth*) *manifold* of dimension n is a set $\mathcal{M}$ and a maximal family of *injective* mappings $\varphi_i : U_i \subset \mathbf{R}^n \to \mathcal{M}$ of open sets U_i of $\mathbb{R}^n$ into $\mathcal{M}$ such that:

1. $\cup_i \varphi_i(U_i) = \mathcal{M}$
2. for any pair i, j with $\varphi_i(U_i) \cap \varphi_j(U_j) = W \neq \phi$, the sets $\varphi_i^{-1}(W)$ and $\varphi_j^{-1}(W)$ are open sets in $\mathbb{R}^n$ and the mappings $\varphi_j^{-1} \circ \varphi_i$ are differentiable, where $\circ$ denotes function composition.

3. The family $\{(U_i, \varphi_i)\}$ is maximal relative to the conditions (1) and (2).

Intuitively, a differentiable (smooth) manifold $\mathcal{M}$ is a topological space that is locally similar to Euclidean space and has a globally defined differential structure.

Tangent space ($T_p\mathcal{M}$). The *tangent space* at $p \in \mathcal{M}$ is a vector space which consists of tangent vectors of *all* possible curves passing through p.

Tangent bundle ($T\mathcal{M}$). The *tangent bundle* of $\mathcal{M}$ is the disjoint union of tangent spaces at all points of $\mathcal{M}$, $T\mathcal{M} = \coprod_{p \in \mathcal{M}} T_p\mathcal{M}$. The tangent bundle is equipped with a natural *projection map* $\pi : T\mathcal{M} \to \mathcal{M}$.

Riemannian manifold. A *Riemannian manifold* is equipped with a smoothly varying metric (inner product), the *Riemannian metric*.

Various classical geometric notions, e.g., the angle between two curves or the length of a curve, can be extended to manifold spaces.

Geodesic curves. A geodesic curve on a Riemannian manifold is the locally shortest (distance-minimizing) curve. These are analogous to straight lines in Euclidean space and a main object to generalize linear models to Riemannian manifolds (as we will in Chapter 3).

Geodesic distance. The *geodesic distance* between two points on $\mathcal{M}$ is the length of the shortest *geodesic* curve connecting the two points. More generally, distance between two points on Riemannian manifolds is defined by the infimum of the length of all differentiable curves connecting the two points. Let γ be a continuously differentiable curve $\gamma : [a, b] \to \mathcal{M}$ between p and q in $\mathcal{M}$ and g be a metric tensor in $\mathcal{M}$. Then, formally, the distance between p and q is defined as

$$\mathrm{d}(p, q) := \inf_{\gamma} \int_a^b \sqrt{g_{\gamma}(t)(\dot{\gamma}(t), \dot{\gamma}(t))} dt \tag{2.18}$$

where $\gamma(a) = p$ and $\gamma(b) = q$, and $\dot{\gamma}$ is the derivative or rate of change of γ.

Exponential map . An exponential map is a map from a tangent space $T_p\mathcal{M}$ to $\mathcal{M}$, which is usually locally defined due to the existence and uniqueness of an ordinary differential equation for the map. The geodesic curve from y_i to y_j can be

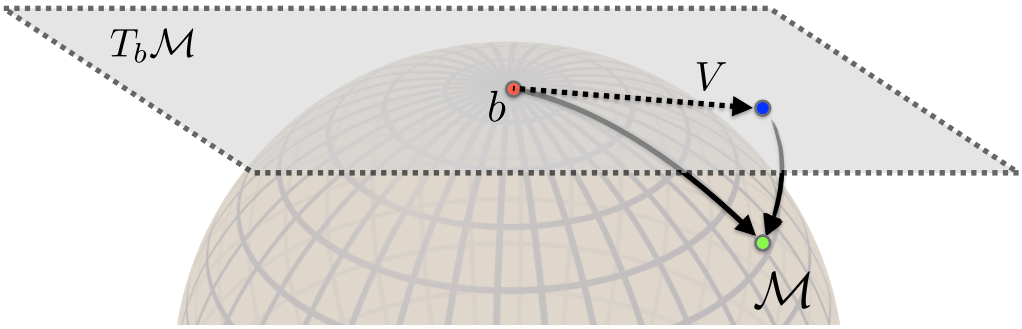


Figure 2.2: From a basepoint b, we can identify the tangent space $T_b\mathcal{M}$, use a direction $V \in T_b\mathcal{M}$ to move using the tangent map, and use the exponential map to project back on to the manifold.

Operation	Euclidean	Riemannian
Subtraction	$\overrightarrow{x_ix_j} = x_j - x_i$	$\overrightarrow{x_ix_j} = \operatorname{Log}(x_i, x_j)$
Addition	$x_i + \overrightarrow{x_jx_k}$	$\operatorname{Exp}(x_i, \overrightarrow{x_jx_k})$
Distance	$\lVert\overrightarrow{x_ix_j}\rVert$	$\lVert\operatorname{Log}(x_i, x_j)\rVert_{x_i}$
Mean	$\sum_{i=1}^{n} \overrightarrow{\bar{x}x_i} = 0$	$\sum_{i=1}^{n} \operatorname{Log}(\bar{x}, x_i) = 0$
Covariance	$\mathbb{E}\left[(x_i - \bar{x})(x_i - \bar{x})^T\right]$	$\mathbb{E}\left[\operatorname{Log}(\bar{x}, x)\operatorname{Log}(\bar{x}, x)^T\right]$

Table 2.1: Basic operations in Euclidean space and Riemannian manifolds.

parameterized by a tangent vector in the tangent space at y_i with an exponential map $\operatorname{Exp}(y_i, \cdot) : T_{y_i}\mathcal{M} \to \mathcal{M}$.

Logarithm map. The inverse of the exponential map is the *logarithm map*, $\operatorname{Log}(y_i, \cdot) : \mathcal{M} \to T_{y_i}\mathcal{M}$. For completeness, Table 2.1 shows corresponding operations in the Euclidean space and Riemannian manifolds. In what follows, when operations are nested, the exponential map and its inverse logarithm map are denoted by $\operatorname{Exp}(p, x)$ and $\operatorname{Log}(p, v)$ respectively, where $p, x \in \mathcal{M}$ and $v \in T_p\mathcal{M}$. They are usually denoted $\operatorname{Exp}_p(x)$ and $\operatorname{Log}_p(v)$ in classical differential geometry literature.

Separate from the above notations, the matrix exponential, i.e, $\exp(X) := \sum \frac{1}{k!}X^k$, where $0! = 1$ and $X^0 = I$ and the matrix logarithm are denoted fully lowercase by as $\exp(\cdot)$ and $\log(\cdot)$, similar to their univariate counterparts. These will be necessary in a specific instance in Chapter 3, but otherwise will refer to the common univariate functions in other contexts.

Intrinsic mean. Let $d(\cdot, \cdot)$ define the distance between two points. The intrinsic (or Karcher) mean is the minimizer to

$$\bar{y} = \arg\min_{y \in \mathcal{M}} \sum_{i=1}^{N} d(y, y_i)^2, \tag{2.19}$$

which may be an arithmetic, geometric or harmonic mean depending on $d(\cdot, \cdot)$. A Karcher mean is a local minimum to (2.19), and a global minimum is referred to as a Fréchet mean. On manifolds, the Karcher mean satisfies $\sum_{i=1}^{N} \operatorname{Log}_{\bar{y}} y_i = 0$. This identity implies the first order necessary condition of (2.19), i.e., $\bar{y}$ is a local minimum with a zero norm gradient (Karcher, 1977). In general, on manifolds, the existence and uniqueness of the Karcher mean is not guaranteed unless we assume, for uniqueness, that the data is in a small neighborhood.

Algorithm 1: Karcher Mean on Manifolds

```
Input: $y_1, \dots, y_N \in \mathcal{M}$, $\alpha$
Output: $\bar{y} \in \mathcal{M}$
while $\| \sum_{i=1}^{N} \operatorname{Log}(\bar{y}_k, y_i) \| > \epsilon$ do
    $\Delta\bar{y} = \frac{\alpha}{N} \sum_{i=1}^{N} \operatorname{Log}(\bar{y}_k, y_i)$;
    $\bar{y}_{k+1} = \operatorname{Exp}(\bar{y}_k, \Delta\bar{y})$
end
```

Parallel transport. Let $\mathcal{M}$ be a differentiable manifold with an affine connection ∇ and I be an open interval. Let $c : I \to \mathcal{M}$ be a differentiable curve in $\mathcal{M}$ and let V_0 be a tangent vector in $T_{c(t_0)}\mathcal{M}$, where $t_0 \in I$. Then, there exists a unique parallel vector field V along c, such that $V(t_0) = V_0$. Here, $V(t)$ is called the *parallel transport* of $V(t_0)$ along c.

Geometry of SPD manifolds

As mentioned above, covariance matrices are symmetric positive definite matrices. Here we focus our discussion to the above operations specific to SPD matrices.

Let SPD(n) be a manifold for symmetric positive definite matrices of size $n \times n$. This forms a quotient space $GL(n)/O(n)$, where $GL(n)$ denotes the general linear group (the group of $(n \times n)$ nonsingular matrices) and $O(n)$ is the orthogonal group (the group of $(n \times n)$ orthogonal matrices). Here, the tangent space $T_p\mathcal{M}$ is the space of symmetric matrices of dimension $(n+1)n/2$.

The inner product of two tangent vectors $u, v \in T_p\mathcal{M}$ is given by

$$\langle u, v\rangle_p = \mathrm{tr}(p^{-1/2}up^{-1}vp^{-1/2}) \tag{2.20}$$

This plays the role of the Fisher-Rao metric in the statistical model of multivariate distributions. The geodesic distance is $d(p,q)^2 = \mathrm{tr}(\log^2(p^{-1/2}qp^{-1/2}))$.

The exponential maps and logarithm maps are given as

$$\mathrm{Exp}(p,v) = p^{1/2}\exp(p^{-1/2}vp^{-1/2})p^{1/2}, \quad \mathrm{Log}(p,q) = p^{1/2}\log(p^{-1/2}qp^{-1/2})p^{1/2}. \tag{2.21}$$

Let p, q be in SPD(n) and a tangent vector $w \in T_p\mathcal{M}$, the tangent vector in $T_q\mathcal{M}$ which is the parallel transport of w along the shortest geodesic from p to q is given by

$$\begin{aligned}&\Gamma_{p\to q}(w) = p^{1/2}rp^{-1/2}wp^{-1/2}rp^{1/2}\\ &\text{where } r = \exp\left(p^{-1/2}\frac{v}{2}p^{-1/2}\right) \text{ and } v = \mathrm{Log}(p,q)\end{aligned} \tag{2.22}$$

Other spaces of interest. Orthogonal matrices of fixed size and rank also form a manifold, the (compact) **Stiefel Manifold**: St$(p,n) = \left\{Y \in \mathbb{R}^{n\times p} | Y^TY = I_p,\ p \leq n\right\}$. An arbitrary $X \in \mathbb{R}^{n\times p}$ matrix can be projected onto the Stiefel manifold St(p,n) using $X \mapsto UV^T$ where $X = U\Sigma V^T$ is the (thin) singular value decomposition of X. We will use this fact in Chapter 4.

Tensors and their Geometry

Let $T \in \mathbb{R}^{n_1\times\cdots\times n_d}$ be a d-dimensional array, or tensor, with each mode having length n_i. To store a full rank tensor, n^d storage would be required. A number of tensor factorizations have been developed to reduce this storage cost. The CANDECOMP/PARAFAC (CP) decomposition (Harshman, 1970; Carroll and Chang, 1970) reduces the storage to $O(dnr)$, reducing the tensor to a sum of R rank-1 weighted outer products:

$$T^{CP} = \sum_{r=1}^{R} \lambda_r x_1^r \otimes \cdots \otimes x_d^r \tag{2.23}$$

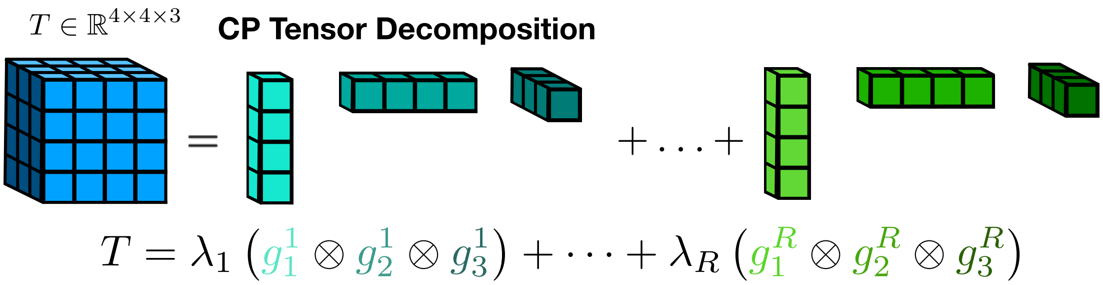

$T \in \mathbb{R}^{4\times4\times3}$
CP Tensor Decomposition
=
$+\ldots+$
$T = \lambda_1 \left(g_1^1 \otimes g_2^1 \otimes g_3^1\right) + \cdots + \lambda_R \left(g_1^R \otimes g_2^R \otimes g_3^R\right)$

Figure 2.3: CP-style decomposition of an arbitrary tensor.

where vectors $x_i^r \in \mathbb{R}^{n_i}$. Finding the exact CP-rank r is NP-hard, however. An alternative decomposition decomposes the tensor into sets of matrices and one smaller "core" tensor $\mathcal{T}$. Generalizing the above,

$$T^{Tucker} = \mathcal{T} \otimes_1 G_1 \ldots \otimes_d G_d \tag{2.24}$$

where $\mathcal{T} \in \mathbb{R}^{k_1,\ldots,k_d}$ and the outer products are taken along the corresponding dimension. The size of $\mathcal{T}$ is defined as the *Tucker rank*, and when $\mathcal{X} = \mathcal{X}^{Tucker}$, is analogous to the number of nonzero eigenvalues for a matrix (tensor of dimension 2). In this way, it is also considered a *higher-order singular value decomposition*, and algorithms exist for computing it directly. Unfortunately its space complexity is $O(dnr + r^d)$, reasonable for lower-order tensors but unsuitable as the order grows. Hierarchical tensor methods have also proven to be effective in tensor compression (Cohen et al., 2016; Cohen and Shashua, 2016), and have led to a newer construction with interesting properties.

A more recent decomposition, the *Tensor Train* decomposition (TT) (Oseledets, 2011), defines an element of the tensor as

$$T(x_1, \ldots, x_d) = A_1(x_1) \cdots A_d(x_d) \tag{2.25}$$

where $x_i \in \{1, \ldots, n_i\}$, and $A_i(x_i) \in \mathbb{R}^{r_{i-1}\times r_i}$ for each $i \in \{1, \ldots, d\}$ are called the *cores* of the tensor train, with $r_0 = r_d = 1$. Equivalently, the full tensor is written as:

$$T = \sum_{k_0=1}^{r_0} \cdots \sum_{k_d=1}^{r_d} A_1(k_0, :, k_1) \otimes \cdots \otimes A_d(k_{d-1}, :, k_d) \tag{2.26}$$

where $A_i \in \mathbb{R}^{r_{i-1}\times n_i \times r_i}$. This format requires $O(dnr^2)$ storage, but has two major

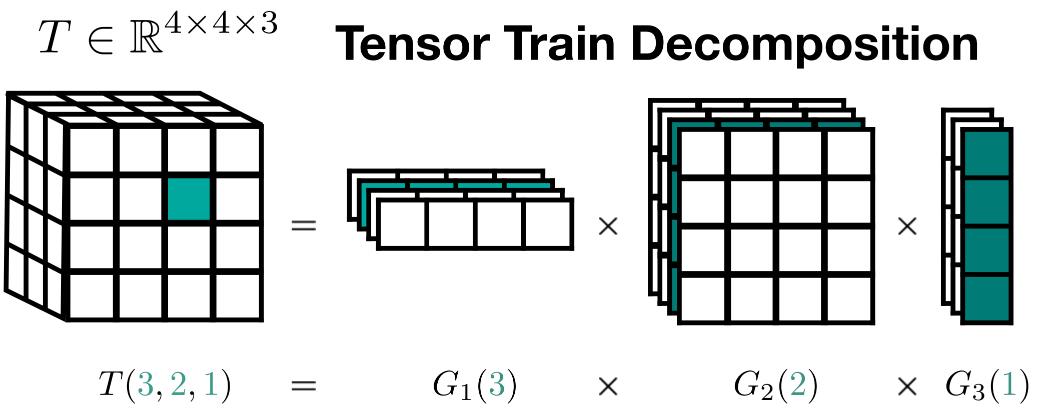

$T \in \mathbb{R}^{4\times4\times3}$
Tensor Train Decomposition
=
×
×
$T(3, 2, 1)$
=
$G_1(3)$
×
$G_2(2)$
×
$G_3(1)$

Figure 2.4: Visualization of a tensor train decomposition.

advantages over the CP format. First, finding the TT-rank (the smallest set of r_i's that satisfy the decomposition with equality) of any arbitrary tensor is tractable, and as such all tensors can be efficiently rewritten in the TT format. Second, projecting arbitrary tensors onto the TT format of a fixed rank requires only a set of QR and singular value decompositions (Oseledets, 2011). This projection, *TT-rounding*, additionally allows for a given TT tensor of some rank to be projected onto the space of TTs with lower rank, and requires $O(dr^3)$ computational complexity. Separately, specific tensor train constructions have recently been identified as forms of general recurrent networks (Khrulkov et al., 2019). We denote a tensor operator $\mathcal{G}$ as a grouping of tensor modes into an "input" and "output" list, such that $\mathcal{G} \in \mathbb{R}^{(n_1^{in},\times\ldots\times,n_d^{in})\times(n_1^{out},\times\ldots\times,n_d^{out})}$. This operator $\mathcal{G}$ can be seen as the TT representation of a matrix $W \in \mathbb{R}^{(n_1^{i}\cdots n_d^{i})\times(n_1^{o}\cdots n_d^{o})}$. In Novikov et al. (2015), authors use this formulation to directly compress the weight layers in neural networks. Cores in the operator are indexed by both an input and output index, i.e., $A_i(x_i, y_i) \in \mathbb{R}^{r_{i-1}\times r_i}$, where $x_i \in [1, \ldots, n_i^{in}], y_i \in [1, \ldots, n_i^{out}]$.

Common operations upon tensor trains require *matricizing* the cores of the TT format. Here, we define the left matricization of core $A_i(x_i)$ as

$$A_i(x_i) = A_i^L \in \mathbb{R}^{r_{i-1}n_i\times r_i} \tag{2.27}$$

and the right matricization similarly. Other desirable operations are fully supported by the format as well, including most linear algebra operations such as summations, multiplications, Frobenius norms, and decomposition into the format via an iterative higher-order SVD procedure. For brevity in this thesis, we refer interested readers to the original tensor train proposal and citations above.

Differential Geometry of Tensor Trains

Tensor trains with fixed TT-ranks form a Riemannian submanifold of $\mathbb{R}^{n_1 \times \cdots \times n_d}$ (Lubich et al., 2015; Holtz et al., 2012):

$$\mathcal{M}_r := \{T \in \mathbb{R}^{n_1 \times \cdots \times n_d} \text{ with TT-ranks } r_0, \ldots, r_d\} \tag{2.28}$$

Optimizing a function with respect to a Riemannian manifold-valued variable amounts to computing a free derivative in the ambient space, projecting the gradient to the tangent space of the current iterate, and using the (retraction) exponential map to compute the next iterate. The authors in Novikov et al. (2017) use this procedure to more effectively learn a model of all exponentially many interactions in a linear model.

2.4 Deep Networks, Optimization, and Objectives

The form of the function f_θ in (1.2) is critical in determining both the types of optimization methods that may identify a solution, and the particular minimizer identified. In the learning methods that follow, f will typically take the form of a deep neural network. The advantages and successes of deep neural networks rely heavily on their ease of optimization: the *computation graph* that underlies the neural network allows for gradients to be computed by parts and accumulated via the chain rule.

Consider a simple function $f_\theta(x)$ that is defined as a linear combination of some parameters $\theta := w, w \in \mathbb{R}^d$ with $x \in \mathbb{R}^d$ followed by a differentiable, nonlinear scalar *activation* function $a(\cdot)$:

$$f_\theta(x) := a(w \cdot x) \tag{2.29}$$

If we have some estimate of the parameters $\theta := w$, then the gradient of the full network with respect to those parameters is

$$\frac{df}{d\theta} = \frac{df}{da}\frac{da}{dw} \tag{2.30}$$

where df/da is the (known) derivative of the *activation* function, and da/dw is exactly x, the derivative of a linear function. With a direction of descent, we can update the parameters via some update to minimize the functional f of interest:

$$\theta_{t+1} = \theta_t + g_x(\theta_t) \tag{2.31}$$

where $g(\cdot)$ is some function of the full derivative $g_x(\cdot) := g(\nabla f_{\theta_t}(x))$, and θ_t are the current parameter estimates. Optimization proceeds and terminates when a certain amount of iterations t have completed, or some stopping criterion has been reached, typically that the gradient is small, indicating that a minima has been identified.

In data science and machine learning applications, we typically do not have a fixed x, but rather a dataset $X := \{x_i\}_{i=1}^n$. If the dataset is small, we may be able to still compute an update as in (2.31). But this is infeasible when we have thousands, hundreds of thousands, or millions of samples. In this case, stochastic gradient descent (SGD) is used. Gradient updates in SGD proceed by taking a single sample and computing (2.31). *Training* of θ follows by iteratively updating the parameters over full passes of the dataset. When feasible, mini-batches of samples can be used instead of a single sample, and in both cases convergence and convergence rates have been shown to be reasonable (Hardt et al., 2016).

Hessians. Uninformed gradient updates are generally preferred for their speed and ease of computation. However, additional information in the form of the *Hessian* can lead to faster convergence as well as a number of theoretical properties and guarantees. Consider the function at a critical point θ^*. The Taylor expansion of the function at that point is

$$f(\theta) = f(\theta^*) + \nabla f(\theta^*)^\top f(\theta - \theta^*) + \frac{1}{2}(\theta - \theta^*)^\top H(\theta^*)(\theta - \theta^*) + \ldots \tag{2.32}$$

Where $H(\theta^*)$ is the Hessian matrix at the point θ^*. With no additional terms, this second-order approximation provides information about the local curvature of the function near the critical point, allowing a scaling of the gradient that can use this local curvature to inform optimization:

$$\theta_{t+1} = \theta_t + H(\theta_t)^{-1} g(\theta_t) \tag{2.33}$$

These Newton updates are typically infeasible in most machine learning applications with high-dimensional parameter spaces, where the complexity of the actual function or the maximal moment is unknown. In some cases, Hessian approximations, or its eigenspectrum can be efficiently computed, and as we will see this can lead to practical measures that lead to new algorithms and guarantees.

For more on these ideas, and a formal treatment with respect to general optimization, see Wright et al. (1999).

These formulations and SGD updates generalize to extremely large and complex stacks of operations, and are what have enabled the enormous success and ubiquity of learning methods. While “fully-connected" layers, such as in (2.29) are the simple in their form, they have been proven to be sufficient in large capacity to serve as *universal function approximators* (Cybenko, 1989). In this form however, capacity guarantees require the number of parameters (dimension of w) to grow exponentially: infeasible in practice. If the size is misspecified, training can lead to parameter settings that are provably optima at that level, but fail to sufficiently capture the true problem complexity. These problems extend to stacks of “fully-connected" layers as well: nonlinearity through multiplication and activations is effective, but can lead to poor training time results due to convergence to local minima.

The final minima identified can vary significantly based on the particular form of the function f_θ, and, in the case where the function f is not *convex*, it can also depend on the estimate θ_0 at initialization. This detail has been used to suggest that large, complex neural networks may contain sub-networks that can be trained in isolation to perform a task with high accuracy, even when randomly initialized. The “lottery ticket hypothesis" states a small, sparse sub-network can be trained to perform just as well as the larger network, but with fewer parameters and less computation (Frankle and Carbin, 2019).

A wide variety of neural network *architectures* have been proposed depending on the type of data and learning task. Some of the most commonly used architectures include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Transformers, Autoencoders, and Generative Adversarial Networks (GANs). CNNs, mainly used in image recognition and processing tasks, use convolutions designed to process grid-like imaging data. RNNs are designed to handle sequential data such as time series or language, and have a memory-like mechanism that allows information to persist in hidden state vectors. Transformers, now the de-facto method in Natural

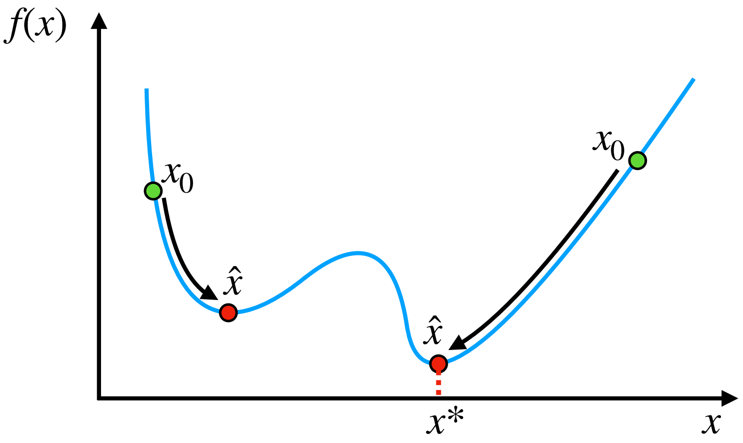


Figure 2.5: Nonconvex functions $f(x)$ optimized using variations of gradient descent can lead to different (and potentially non-global) optima ($\hat{x}$).

Language Processing (NLP), make use of a self-attention mechanism, weighing the importance of different input tokens for eventual prediction (Vaswani et al., 2017). Autoencoders and Generative Adversarial Networks (GANs) fall under the category of generative models, trained to generate outputs that resembles real data. Newer generative models built on diffusion methods have been able to produce photorealistic images based only on captions input by the user (Rombach et al., 2022). All of these methods come with their own idiosyncracies in model capacity and practical methods for efficient training (LeCun et al., 2015).

With these varying architectures has also come the field of *architecture search*, identifying the hyperparameters and layer types that would lead to the best downstream performance with the most efficient full architecture. While computationally expensive, techniques such as reinforcement learning, evolutionary algorithms, gradient-based methods, and even greedy approaches have been used to identify state-of-the-art networks (Elsken et al., 2019).

Losses and Probability Measures

The methods for optimization above lend themselves to typically arbitrary functions. As described in the Introduction, typically a *loss function* is defined to measure the disparity between the prediction or output of a model and the target of interest. Optimizer flexibility has led to the development and design of loss functions that suit particular tasks, or those that correspond more directly to practitioners' high-level goals. Building on the classical mean-squared error, $\ell(f_\theta(x), y) := (y - f_\theta(x))^2$, methods have extended to information-based schemes such as cross-entropy, as well as incorporating classical *regularization* schemes to push solutions towards desirable minima. These typically take the form of additive terms penalizing large norms over weights, where the norm chosen corresponds to the choice of prior assumed by the user.

Following the information-based schema, significant work has been done on probabilistic forms of losses, treating the input and output spaces of models as distributions. Particularly useful for generative models, f-divergences have been studied as a general form of distances between probability distributions that can be effectively minimized to train in such a way that model outputs come from a maximum-entropy distribution with respect to the original training data (Nowozin et al., 2016). Measures such as KL-divergence, mutual information, maximum-mean discrepency, and others all fall within this framing.

Optimal Transport

Of particular interest in this thesis is the *Wasserstein* distance, or traditionally known as the Earth Mover's distance. Recent developments in GANs (Arjovsky et al., 2017) have demonstrated the Wasserstein metric is typically more stable compared to other measures and avoids *mode collapse*, sharp local minima with low variation. Interestingly this formulation can be viewed as an approximation of the *optimal transport* problem.

Optimal transport refers to the problem of finding a transportation plan that minimizes some cost of transforming one probability distribution to another. Developments in Riemannaian geometry and measure theory have led to a general formulation.

Definition 2.4 (Monge-Kantorovich Problem (Kantorovich, 2006))**.** *Let μ, ν be probability measures over separable metric spaces X and Y. The optimal transportation problem seeks to find a joint measure γ on $X \times Y$ that satsifies*

$$\gamma^* = \mathit{inf}\left\{ \int_{X\times Y} c(x,y)d\gamma(x,y) \middle| \gamma \in \Gamma(\mu,\nu) \right\} \tag{2.34}$$

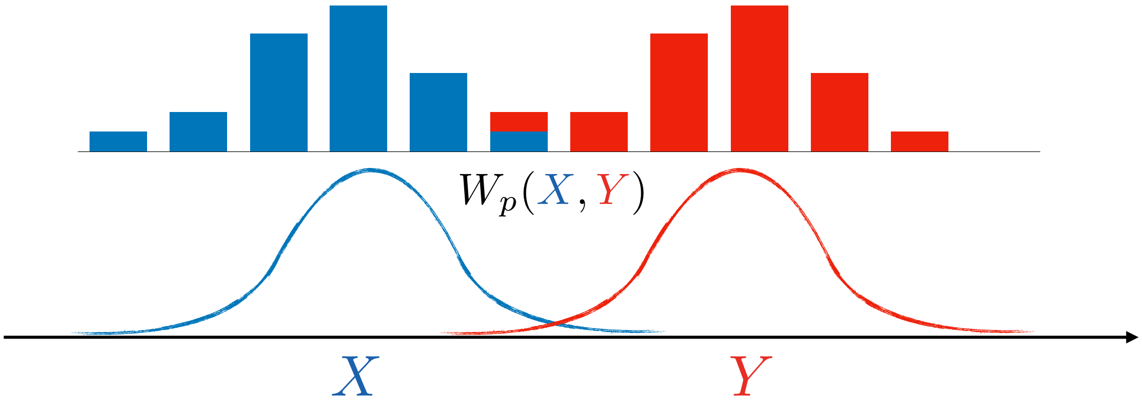


Figure 2.6: 2D Optimal transport measures the distance or cost associated with transforming one discrete (above) or continuous (below) distribution to another.

Where Γ is the space of all probability measures with marginals equal to μ on X and ν on Y. More common in recent machine learning is the *Wasserstein* distance, defined as the p-th distance over the Monge-Kantorovich problem.

Definition 2.5 (Wasserstein metric)**.** *The Wasserstein p-distance is given by:*

$$W_p(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \mathbb{E}_{(x,y)\sim\gamma} d(x, y)^p \right)^{1/p} \tag{2.35}$$

And the discrete analog:

Definition 2.6 (Earth Mover's Distance)**.** *Let p_1 and p_2 be distributions with discrete support of size n, and $x(i, j)$, where $x \in \mathbb{R}^{n\times n}$, denotes the movement of "mass" from $p_1(i)$ to $p_2(j)$. Denote by $c(i, j)$ the cost of moving one unit of mass from $p_1(i)$ to $p_2(j)$. The Earth Mover's Distance (EMD) between p_1 and p_2 is the minimal cost to transform p_1 into p_2. Written as a linear program (LP):*

$$\mathit{min}_{x\in\mathbb{R}_+^{n\times n}} \sum_{i,j} c(i, j)x(i, j) \quad \mathit{s.t.} \quad \sum_j x(i, j) = p_1(i); \; \sum_i x(i, j) = p_2(j), \; (\forall i, j \in [n]). \tag{2.36}$$

Computation of the continuous measures can be straightforward with some assumptions, leading to natural linear programming formulations akin to the EMD. However in cases where one wishes to *minimize* this distance, practical complexity explodes, even in the discrete formulations commonly found in application. An entropic regularization method introduced in Cuturi (2013) has led to newfound interest, use, and analysis of optimal transport for deep learning applications. In Chapter 6 we will expand upon these ideas to the case where we have multiple distributions, and wish to minimize the distance among all of them concurrently.

Each chapter in the sequel is defined by a unique intersection of the above ideas, leading to new insights with respect to the subset selection problem of interest. The relevant background will be referenced and refreshed as needed to ease narrative continuity.

Chapter 3

Localizing Group Differences over Covariance Trajectories

Important feature identification has been well studied in classical statistics, and we will start here by adapting those methods directly. We will see how marrying methods from differential geometry and scan statistics enable identifying meaningful features that indicate group differences *over time*, that otherwise would be undetectable given direct applications of classical approaches. We describe theoretical developments on graphical hypothesis testing, enabling the identification of scientifically important and interesting group differences where classical hypothesis testing both does not identify meaningful features, and does not identify unique differences among known disparate groups of individuals. The work presented in this chapter was published as a journal article in the Quarterly of Applied Mathematics (Mehta et al., 2019b).

3.1 Introduction

As described in the previous chapter, a number of methods have been extensively studied for the sparse recovery of graphical models with various assumptions on collected data. Often, data come from two (or more) disparate sources or multiple timepoints. Proposals in the literature have described strategies for linking the sparsity patterns of multiple graphical models, e.g., using a fused lasso penalty (Danaher et al., 2014; Yang et al., 2015). Observe that if the data sources correspond to *longitudinal* acquisitions, we should expect the 'structure' to gradually evolve. Several authors have offered generalizations to address this problem: Zhou et al. (2010) removes the assumption that each graph is independent and structurally 'close'. Instead, Zhou et al. (2010) can be thought of as a growth model (McArdle and Bell, 2000) defined on these structures: they show how non-identically distributed graphs can be learned over time. The nonparametric procedure in Qiu et al. (2015) extends these ideas to handle multiple sources, each with multiple samples.

The ideas in the literature so far to "couple" multiple graphical model estimation modules are mostly nonparametric. While such formulations offer benefits, in many estimation problems, parametric models may be more convenient for downstream statistical analysis, particularly for hypothesis testing (Hardle and Mammen, 1993; Geer, 2000; Roehrig, 1988). Algorithms for *parametric estimation* of temporal or coupled Gaussian graphical models have not yet been heavily studied. This will involve parameterizing *trends* in the highly structured nature of the 'response' variable (SPD matrices). More recently, we find that parametric formulations for manifold-valued data *have* been proposed (Kim et al., 2014; Cornea et al., 2016). Because SPD matrices form a Riemannian manifold, algorithms that estimate a parametric model respecting the underlying Riemannian metric are more suitable in many applications, as opposed to assuming a Euclidean metric on positively or negatively curved spaces (Xie et al., 2010; Fletcher and Joshi, 2007; Jayasumana et al., 2013). We will make a few simple modifications (for efficiency purposes) to such algorithms and make use of the estimated parameters for follow-up analysis.

Finding Group-wise Differences. Assuming that we have a black-box procedure to estimate a parametric model on the SPD manifold available, in many tasks, such an estimation is merely a segue to other analyses designed to answer scientifically meaningful questions. For example, we are often interested in asking whether the temporally coupled model estimated using the procedure above differs in meaningful ways *across* groups induced by a stratification or dichotomous variable (e.g., gender or disease). For instance, is the 'slope' in a structured response space statistically different across education level or body mass index? While existing work in graphical model estimation is mature, the literature describing hypothesis tests in this regime (Städler and Mukherjee, 2012; Belilovsky et al., 2015) is relatively nascent. Given that such questions are simpler to answer with alternative schemes (with assumptions on the distributional properties of the data), e.g., structural equation modeling, latent growth models and so on (Ullman and Bentler, 2003; McArdle and Bell, 2000), it seems that the unavailability of such tools is limiting the adoption of such ideas in a broader cross-section of science. Here we will seek to address this gap.

Needles in Temporal Haystacks. If we temporarily set aside the potential value of a hypothesis test framework for temporal trajectories in graphical models, we see that from an operational viewpoint, such procedures are most effective when a practitioner already has a precise scientific question in mind. In reality, however, many data analysis tools are deployed for exploratory analyses to inform an investigator as to which questions to ask (which subsets to test). Being able to "localize" which parts of the model are different across groups over the entire time window can be very valuable, and it is this **feature identification** that we will study here. This ability actually benefits statistical power as well. Notice that when the stratified groups are not very different to begin with, e.g., healthy individuals with presence or absence of a genetic mutation, the effect sizes are likely to be poor. Here, while the trends identified on the *full* precision matrix may still be different (i.e., there may be a *real* signal associated with a grouping variable), they may not be strong enough to survive significance thresholds. Ideally, what we need here are analogs of widely used "scan statistics" for our hypothesis testing formulations for temporal graphical models — to identify which parts of the signal are promising. Then, even if only a **small subset of features** were different across groups over all time, we may be able to identify these differential effects efficiently. This benefits Type 2 error, provides a practical turnkey product for an experimental scientist, and makes up the key technical results of this chapter.

Foundations of our work can be traced back to fundamental developments made by Ulf Grenander across fields of study. Early work with Rosenblatt on the analysis of stochastic processes and time series first brought to light the fundamental issues of linear modeling in Euclidean space, and demonstrated that in many cases it is necessary to develop methods that take explicit advantage of the inherent structure within data (Grenander and Rosenblatt, 1957). Further pioneering work on the statistical analysis on Lie groups (Grenander, 2008) provides the basis of the Riemannian statistics mentioned above. Modern hypothesis testing of these structured, manifold-valued data in image analysis is built upon the his joint work (Grenander and Miller, 1998). Here, we marry modern developments in these areas, using recent strides in linear model fitting on manifolds and statistical testing of structured data to develop groupwise testing procedures for longitudinal covariances. Concurrent to this work, Su et al. (2014) and Zhang et al. (2018) have developed similar methods of analyzing the statistical properties of trajectories on the $\mathrm{SPD}(n)$ manifold via the transported square-root vector field. While here we focus on a simple approach to enable localization, these developments can be incorporated into this construction.

Contributions. Briefly, we provide **(i)** a simple and efficient parametric procedure for modeling temporally evolving graphical models, **(ii)** a hypothesis test for identifying differences between group-wise estimated models, and **(iii)** a scan algorithm to identify *those subsets of the features which contribute to the group-wise differences*. Together, these ideas offer a framework for identifying group-wise differences in temporally coupled graphical models. From the experimental perspective, we find scientifically plausible results on a unique longitudinally tracked cohort of middle-aged (and young elderly) persons at risk for Alzheimer's disease due to family history, but who are otherwise completely cognitively healthy.

The rest of this chapter is organized as follows. In Section 3.2 we present an efficient manifold regression procedure for modeling covariance trajectories, which serves as a blackbox module in our hypothesis testing framework. In Section 3.3, we define our main hypothesis test for group difference analysis over covariance trajectories. In Section 3.4, we present a set of technical results describing our localization procedure based on scan statistics, as well as derive suitable size corrections to compare across feature subsets. Sections 3.5, 3.6, and 3.8 conclude this chapter with empirical evaluations of our model on synthetic data, various types of demographics/behavior data collected longitudinally in the United States from publicly available resources, and finally, our analysis on a unique longitudinal dataset (followed from 2001 to 2017) from a preclinical Alzheimer's disease study involving approximately 1500 individuals.

3.2 Characterizing Covariance Trajectories

Our main statistical testing framework, to be described shortly, needs an efficient means for calculating a "trajectory" of the feature-by-feature interaction graphs over time for the given longitudinal data. We now describe a scheme which offers this capability. Let $X_t \in \mathbb{R}^{n_t,p}$ be the design matrix of all n_t samples at time t, where

$t \in \{1, \ldots, T\}$, and T is the total number of distinct timepoints. We wish to capture the trends in the relationships between the features as a function of t. To evaluate the groupwise differences in changes of such interactions, we make use of the fact that these interactions are commonly captured by correlation or conditional independence, represented by the covariance matrix (with normalized features) and the precision matrix (the inverse of covariance matrix).

Here we simply use the covariance matrix for each timepoint t to denote the interaction between features, $C_t = cov(X_t)$. Our goal now is to estimate the parameters of the function, $t \to C_t$. We may vectorize the covariance matrix and apply a linear model; its parameters will give the trajectory in "vectorized covariance space" as we scan through t. But these predictions are *not* guaranteed to be valid SPD matrices and even if a projection is performed to obtain a covariance estimate, distortions introduced by the process may be significant (Fletcher, 2013). It is well known that classical vector space models tend to be suboptimal in the manifold setting (covariance matrices live on the SPD manifold) since they use Euclidean metrics which are defined in the ambient space. For manifold-valued data, Riemannian metrics are shown to be superior in many applications (Fletcher and Joshi, 2007; Banerjee et al., 2015; Jayasumana et al., 2013; Tuzel et al., 2007), and are increasingly being deployed in machine learning and statistics. We will utilize an appropriate statistical model informed by the manifold-structure of the data and then derive a hypothesis testing procedure to detect groupwise difference in the changes of interactions between features in longitudinal analysis.

Riemannian Manifold Regression

Several regression models for manifold-valued data have been proposed, a majority of which are nonparametric (Jayasumana et al., 2013; Banerjee et al., 2015). Because of the longitudinal nature of our dataset (and recruitment considerations in neuroimaging studies), sample sizes do not exceed a few hundred participants (typically much smaller). We have found that generally, in this regime, parametric methods are better suited and also offer other benefits for downstream applications. Next, we will give a simple parametric model for this problem. Let x and y be vectors in $\mathbb{R}^p$ and $\mathbb{R}^{p'}$ respectively.

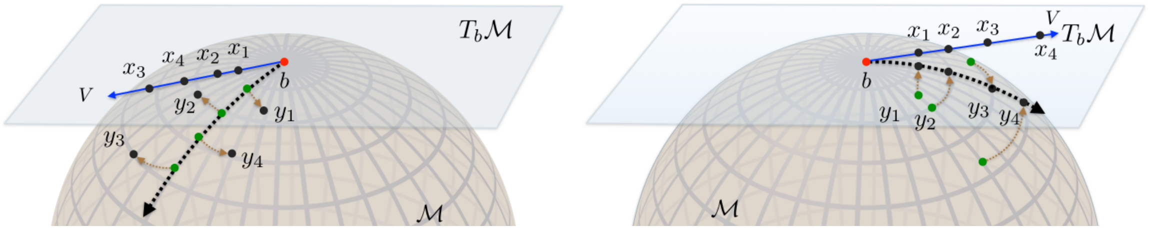


Figure 3.1: Group-wise MMGLM: The left and right figures represent two linear models on the $\mathrm{SPD}(p)$ manifold. Points x_i in the tangent space are our covariate or predictor, and points y_i in the manifold space represent $\mathrm{SPD}(p)$ matrices. In our regression setting, we wish to minimize the error (brown curves) between the estimation and the sample points. Because each linear model has a different base point, the trajectories cannot be directly compared as in the Euclidean setting.

Definition 3.1. (*Standard GLM.*) *The Euclidean multivariate multilinear model is*

$$y = \beta^0 + \beta^1 x^1 + \beta^2 x^2 + \ldots + \beta^p x^p + \epsilon \tag{3.1}$$

where β^0, β^i and the error ϵ are in $\mathbb{R}^p$ and $x = [x^1 \ldots x^p]^T$ are the predictor variables.

Henceforth, we will use the terms *covariate* and *predictor* interchangeably to describe those specific features we wish to control for in our model (e.g., time-points in our experiments). For manifold-valued data, we adapt the formulation proposed by Kim et al. (2014).

Definition 3.2. *The Manifold Multivariate General Linear Model (MMGLM) is defined as*

$$\min_{b \in \mathcal{M}, \forall j, V^j \in T_b\mathcal{M}} \quad \frac{1}{2} \sum_{i=1}^{N} d(\mathrm{Exp}(b, \boldsymbol{V} x_i), y_i)^2, \tag{3.2}$$

where $\boldsymbol{V} x_i := \sum_{j=1}^{n} V^j x_i^j$, and $d(\cdot, \cdot)$ is the geodesic distance between $\hat{y}_i := \mathrm{Exp}(b, \boldsymbol{V} x_i)$ and y_i.

This formulation generalizes (3.1), by replacing the intercept β^0 and each vector β^j for a covariate with a base point $b \in \mathcal{M}$ and a geodesic basis $V^j \in T_b\mathcal{M}$ respectively. The geodesic basis V^j at b parameterizes a geodesic curve $\mathrm{Exp}(b, V^j x^j)$. Intuitively, this model is a "generalized" linear model with the inverse exponential map Exp^{-1} (or logarithm map Log) as a 'link' function (Kim et al., 2014; Cornea et al., 2016). When the covariate/predictors are univariate, we will obtain a single geodesic curve, modeled via the so-called Geodesic Regression (Fletcher, 2013; Shi et al., 2009; Zhu et al., 2009; Yuan et al., 2012).

Efficient Estimation of Trajectories

The objective in (3.2), can be solved by both gradient descent (Fletcher, 2013; Kim et al., 2014) and MCMC methods (Cornea et al., 2016). Unfortunately, these schemes can be expensive, especially when the dimension of the manifold is large. Further, if the algorithm needs to be run a large number of times, the computational footprint quickly becomes prohibitive. Motivated by these considerations, we use a so-called log-Euclidean approximate algorithm introduced in Kim et al. (2014) with some adaptations, which requires mild assumptions on the manifold-valued data.

Recall that in classical ordinary least squares (OLS), the regression curve goes through the mean of covariates and response variables, i.e., $y - \bar{y} = \beta(x - \bar{x})$. Similarly, we assume that geodesic curves go through the mean of response variables on the manifold. Then, the base point, or intercept, "b" in (3.2) can be approximated by the *manifold-valued mean of the sample points*, the Karcher mean (Karcher, 1977). The propositions derived from Kim et al. (2014) lead directly to the following.

Proposition 3.3. *Let $\bar{C}$ be the unique Karcher mean of a sufficiently close set of covariance matrices that lie on a curve Ω. Then $\bar{C} \in \Omega$, and for some tangent vector $V \in T_{\bar{C}}\mathcal{M}$ and each C, there exists $x \in \mathbb{R}$ such that $C = \mathrm{Exp}(\bar{C}, Vx)$.*

This allows us to bypass the fairly involved variational procedure to estimate the base point b.

With this approximation of $\hat{b}$ via $\bar{y}$, the remaining variables to optimize are the tangent vectors $\boldsymbol{V}$. We do so by taking advantage of log-Euclidean schemes. Once the base point is established as the Karcher mean, each data point on the manifold is projected into the tangent space at that point: $\mathrm{Log}(\bar{y}, y)$. These "centered" points $\tilde{y}$ are now Euclidean, and if the covariates are centered as well ($\tilde{x}$), a closed form solution exists in the standard form of $\boldsymbol{V} = \tilde{y}\tilde{x}^\top(\tilde{x}\tilde{x}^\top)^{-1}$ (ordinary least squares).

In this setting, it is often assumed that two points y_1, y_2 have a distance defined as $d(y_1, y_2) := \| \mathrm{Log}(y_1, y_2)\|_{y_1} \approx \| \mathrm{Log}(b, y_1) - \mathrm{Log}(b, y_2)\|_b$. However, on SPD manifolds with an affine invariant metric, each tangent space has a different inner product varying as a function of the base point b, i.e., $\langle u, v\rangle_b := \mathrm{tr}(b^{-1/2}ub^{-1}vb^{-1/2})$. This makes comparison of trajectories difficult without moving to tangent bundle formulations.

This issue is discussed in some detail in Muralidharan and Fletcher (2012); Hong et al. (2015). However, note that

Remark 3.4. *When the base point b is the identity I, then the inner product is exactly the Euclidean metric* $\langle u, v\rangle_b := \mathrm{tr}(b^{-1/2}ub^{-1}vb^{-1/2}) = \mathrm{tr}(uv) = \mathrm{tr}(u^T v)$.

This follows from the fact that u and v are symmetric matrices on $\mathrm{SPD}(p)$. We take advantage of this property through *parallel transport*. Specifically, we can bring all of the data to $T_I\mathcal{M}$ which will allow for a meaningful comparison of two tangent vectors from different base points. Similar schemes have been used for projection on submanifolds in Xie et al. (2010) and other problems (Sommer et al., 2014). With a fast algorithm to compute (3.2) available, we can now accurately model longitudinal trajectories of covariances matrices. Our statistical procedure described next simply assumes the availability of some suitable scheme to solve the manifold-regression as defined in (3.2) efficiently and does not depend on particular properties of the foregoing algorithm.

3.3 Test Statistics for SPD(n) Trajectories

With an algorithm to construct a regression model for covariance matrix responses in hand, we can now describe a key component of our contribution: a test statistic which allows addressing the main question of interest: *Is the progression/trajectory of covariance matrices (over time) different across two groups?* In the standard two-sample testing problem, a hypothesis test is set up to check if the parameters of each group are significantly different:

$$H_0 : \theta_1 = \theta_2 \quad vs. \quad H_A : \theta_1 \neq \theta_2 \tag{3.3}$$

Recall that in a general linear model (GLM), when testing for mean group differences, the test parameters are the regression slopes from a standard GLM fit. In our setting, the parameters of interest are the population covariance trajectories estimated from the manifold regression in (3.2), see Figure 3.1. While the trajectories and the slopes are related, note that our parameters are estimated *on the manifold*. Two unique manifold trajectories, when projected as simple multivariate responses in Euclidean space, may not be significantly different under the GLM hypothesis testing framework, as has been observed by Du et al. (2014). Returning to our longitudinal trajectory formulation, we have the following naïve Covariance GLM:

Definition 3.5. *Let $vec(C_{g,t})$ be the vectorized covariance matrix at timepoint t for group $g \in \{1, 2\}$. Then the naïve Covariance GLM is defined as*

$$vec(C_{g,t}) = \beta_g^0 + \beta_g t + \epsilon \tag{3.4}$$

with the slope $\theta = \beta$ in the hypothesis test in (3.3), and $vec(\cdot)$ is the vectorized form of the input matrix.

With this model, hypothesis testing reduces to a simple difference of slopes, which is well-studied in classical statistics literature.

Definition 3.6. (*Seber and Lee, 2003*) *Let $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2$ be the multivariate slopes calculated from estimating (3.4). Then an α-level hypothesis test rejects the null hypothesis $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2$ when $L > \chi_p^2|_{1-\alpha}$, where*

$$L = (\hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_2)\Sigma^{-1}(\hat{\boldsymbol{\beta}}_1 - \hat{\boldsymbol{\beta}}_2) \tag{3.5}$$

Knowing that the response space is structured, i.e., our covariance matrices lie on the SPD manifold, we seek a more appropriate test and corresponding test statistic which adequately captures this knowledge.

Observe that we can directly apply the manifold regression in Section 3.2 to solve for a linear model on the manifold. That is, we construct the *manifold* GLM as

Definition 3.7. *Let $C_{g,t}$ be the covariance matrix at timepoint t for group $g \in \{1, 2\}$. Then the Longitudinal-Covariance GLM (LCGLM) is defined as*

$$C_{g,t} = \text{Exp}(b_g, \boldsymbol{V}_g t) \tag{3.6}$$

with b_g and $\boldsymbol{V}_g$ being the base point and tangent vector respectively, as described in Section 3.2.

But instead of solving $p(p-1)/2$ independent regressions, now we must concurrently solve for the entire manifold-valued response variable. In this case, we cannot directly compare our trajectories because they lie in *different* tangent spaces. To accurately compare two tangent vectors, we must parallel transport both vectors to the same tangent space. Once they are both in the same space, we can construct a simple test statistic for the trajectory difference.

$$L = \|\Gamma_{b_1 \to I} \boldsymbol{V}_1 - \Gamma_{b_2 \to I} \boldsymbol{V}_2\|_I^2 \tag{3.7}$$

Recall that the inner product at the Identity I coincides with the Euclidean metric. This can now be naturally interpreted as a difference of slopes, and together with a standard Euclidean Normal noise assumption yields the following hypothesis test.

Proposition 3.8. *Assume that $\Gamma_{b \to I} \boldsymbol{V}$ is normally distributed $N(0, I)$. Then the statistic defined in (3.7) follows a χ_p^2 distribution with p degrees of freedom, and the threshold test in 3.6 is an α-level hypothesis for the covariate trajectory group difference.*

Proof. The proof of this follows directly from the definition of (3.7). The definition for L can equivalently be written as $(\Gamma_{b_1 \to I} \boldsymbol{V}_1 - \Gamma_{b_2 \to I} \boldsymbol{V}_2)^\top I (\Gamma_{b_1 \to I} \boldsymbol{V}_1 - \Gamma_{b_2 \to I} \boldsymbol{V}_2)$, and if the normal distribution assumption holds, it is equal to (3.5) with $\Sigma = I$. □

Incorporating First-Order Differences

In many real world situations, first-order information in the data is often valuable in identifying group differences. Restricting our analysis to only the second-order interactions, i.e., covariances, may be inefficient (or sub-optimal) when the mean signal difference between groups is large. Our construction easily extends to these cases. Particularly, the *product space* over both means and covariances is in $\mathbb{R}^p \times \mathrm{SPD}(p)$.

Remark 3.9. *The typical GLM on the first order information is defined in the standard Euclidean space. So, computing the regression in the product space $\mathbb{R}^p \times \mathrm{SPD}(p)$ amounts to simply computing the regression on the first and second order statistics (mean and covariance) separately.*

The above statement suggests that by applying the manifold regression to the covariances and the standard regression model for the means, we are directly solving the product space regression problem, incorporating both first and second order statistics. However, in these cases, the statistic defined above in (3.7) does *not* directly take into account the potential difference in means. However, given our Normal noise assumption we can easily invoke the standard Gaussian multivariate likelihood statistic for group differences.

Definition 3.10. *Let $\hat{\mu}_t, \hat{\Sigma}_t$ be the estimated mean and covariance from the standard linear model and our manifold-covariance GLM respectively. Then the Gaussian likelihood of our data X is*

$$P(X|\hat{\mu}, \hat{\Sigma}) = \prod_{t=1}^{T} \prod_{i=1}^{n_t} P(X_t|N(\hat{\mu}_t, \hat{\Sigma}_t)), \tag{3.8}$$

where X_t is the subset of our data collected at timepoint t. Additionally, we can define a standard likelihood ratio test statistic as:

$$L_{prod} = \frac{P(X_1|\hat{\mu}_1, \hat{\Sigma}_1)P(X_2|\hat{\mu}_2, \hat{\Sigma}_2)}{P(X_{1,2}|\hat{\mu}_{1,2}, \hat{\Sigma}_{1,2})} \tag{3.9}$$

This statistic is again χ^2_p-distributed (Seber and Lee, 2003), and an α-level hypothesis test for group difference analysis can be defined in the same way as above. While our manifold regression modeling is focused on the case of centered data (where the mean signal may not be significantly different between the groups), we use the product space construction, wherever appropriate, in experimental evaluations.

3.4 Localizing Group Differences for SPD(n) Trajectories

The above procedure provides a precise mechanism to derive a statistic from the group-wise covariance matrix trajectories. However, when the effect sizes are poor, any scheme operating on the trajectories of the *full covariance matrix* may still fail to identify group differences (as is the case in our experiments). To improve statistical power, localizing the process of computing the trajectories *only to the relevant features* (subset selection) is critical. To this end, we consider the following global hypothesis testing problem

$$H_0 : \forall R, \boldsymbol{\beta}_1^R = \boldsymbol{\beta}_2^R \quad vs. \quad H_1 : \exists R, \boldsymbol{\beta}_1^R \neq \boldsymbol{\beta}_2^R,$$

where $\boldsymbol{\beta}$ denotes the slope and R is the **region of the covariance matrix** *which only includes the relevant features*, see Fig. 3.2. It turns out that by adapting *Scan statistics* (Fan et al., 2012; Arias-Castro et al., 2011), we will be able to exclude the effect of irrelevant regions of the covariance matrix in the calculated trajectories. By extending this concept to graphs, we obtain an algorithm to identify *subsets of features* of the covariance matrix which show group differences that are otherwise unidentifiable, in a statistically rigorous way.

Scan Statistics

Scan statistics are a valuable tool for structured multiple testing. In its simplest form, we can consider a setting where we place a window (or box) over a region R in an image and calculate a local statistic L_R, e.g., an average or a response to a convolution filter. Then, the window can be raster scanned at various locations in the image ($\mathcal{R}$) and the maximum over the set of local statistics is called the scan statistic. Intuitively, if the image is assumed to be a Gaussian random field, we can set up a null hypothesis using a critical value and finding a statistically significant signal (i.e., regions) corresponds to comparing the local region-wise statistic with the critical value. Of course, there is flexibility in terms of specifying properties of the regions as described next.

Definition 3.11. *Let $\mathcal{R}$ be the collection of all possible structured regions, and L_R be some statistic over region R, a structured subset of $\mathcal{R}$. The scan statistic is defined as $L^* = \max_{R \in \mathcal{R}} L_R$.*

Recent results in scan statistics show how *size corrections* can be used to increase detection power in multi-scale analysis with nice guarantees (Walther, 2010; Wang et al., 2016). To utilize these ideas for our hypothesis test, we must extend scan statistics and these size corrections to a graph setting where the graph is induced by a sparse estimation of the precision matrix, e.g., graphical lasso (or any other algorithm of choice) over the features. To do so, structured regions R and a statistic L_R on each region must be defined on the graph. Intuitively, in our case, L_R must capture the “difference” in group-wise covariance trajectories. As we will describe shortly, it is in the context of this statistic where we utilize the LCGLM (3.6), which will be invoked at the *level of individual regions R*, one by one.

Let $G := (\mathcal{V}, E)$ be a graph over the features (represented in the covariance matrix) with vertex set $\mathcal{V}$ (to avoid overlap with tangent vectors) and edge set E. We define the structured region $R \subseteq G$ as a connected subgraph of G corresponding to the selection of those vertices as our feature subset (block of the covariance matrix, see Fig. 3.2). A natural question is whether such an enumeration is tractable if the number of connected subgraphs $\mathcal{R}$ is exponential. It turns out that if we make a mild assumption on the graph, the number of induced regions can be shown to be polynomially bounded. Further, it then naturally provides a *size correction*, the analog for a multiple testing adjustment.

In our motivating application, the group differences we seek to identify will involve a cohesive set of features that will be connected to each other, by definition (large changes in covariances indicate dependent features). Based on this observation, we assume that the true localized subgraph is a "ball" subgraph.

Definition 3.12. *A ball subgraph consists of nodes with a given radius r from a particular node (see Fig. 3.2). The collection of ball subgraphs is defined as*

$$\mathcal{R} = \{B(v, r) : v \in \mathcal{V} \text{ and } r \in \mathbb{N}\} \tag{3.10}$$

where the ball subgraph $B(v, r) := \{v' \in \mathcal{V} : d(v, v') \leq r\}$, and $d(v, v')$ is the minimum length path connecting v and v'.

With this assumption, it can be verified that we now only need to search a polynomially bounded set of regions.

Remark 3.13. *The number of unique ball subgraphs in any graph G is bounded above by $D|\mathcal{V}|$, where D is the diameter (longest chain) of the graph G.*

On these regions (i.e., blocks of covariance matrix), we will invoke LCGLM to provide us a statistic L_R. This is just the difference in slopes of the calculated manifold regression across groups in (3.7). We will iteratively obtain this statistic for distinct regions R and find subgraphs that differ in their trajectories across groups using a size correction for hypothesis tests.

Let us revisit the standard linear model setting and assume that our slopes $\boldsymbol{\beta}_g^R$ correspond to the subset of slopes from features in R, and $\hat{\boldsymbol{\beta}}_g^R$ is an estimate of that slope. In this case, we have the following statistic (see e.g. Seber and Lee (2003)),

$$(\hat{\boldsymbol{\beta}}_1^R - \hat{\boldsymbol{\beta}}_2^R)\Sigma_R^{-1}(\hat{\boldsymbol{\beta}}_1^R - \hat{\boldsymbol{\beta}}_2^R) \sim \chi^2_{|E(R)|}, \tag{3.11}$$

where Σ_R^{-1} is the covariance matrix of $\hat{\boldsymbol{\beta}}_1^R - \hat{\boldsymbol{\beta}}_2^R$, and $E(R)$ denotes the number of edges among vertices in the subgraph R. With a normal noise assumption, this covariance will be identity and the statistic would simply be the ℓ_2-norm difference as in the classical analysis. To make the statistics comparable across *different sizes*, we use the

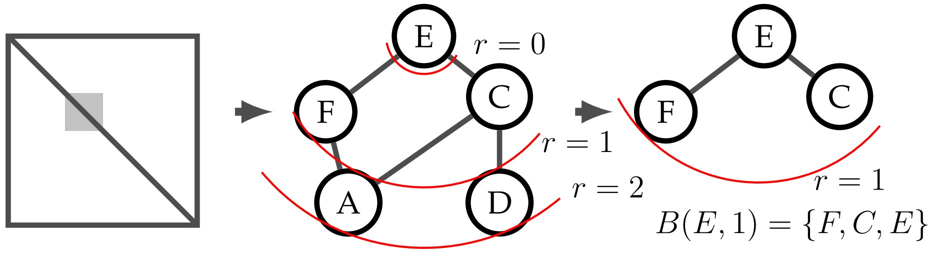


Figure 3.2: (left) A region of the sparse precision matrix, (center) The corresponding subgraph of that region, along with balls of varying radius from the root node E, (right) The ball subgraph constructed with $r = 1$. These subgraphs with bounded radius act as the structured regions on which scan statistics can be applied.

standardized version of a $\chi^2_{|E(R)|}$ distribution,

$$L_R = \frac{(\hat{\boldsymbol{\beta}}_1^R - \hat{\boldsymbol{\beta}}_2^R)\Sigma_R^{-1}(\hat{\boldsymbol{\beta}}_1^R - \hat{\boldsymbol{\beta}}_2^R) - E(R)}{\sqrt{E(R)}}. \tag{3.12}$$

We can extend this analysis to our manifold setting.

Definition 3.14. *For a given structured region R, the region-based LCGLM is written as*

$$(b_g^R, \boldsymbol{V}_g^R) = \arg \min_{(b^R, \boldsymbol{V}^R) \in T\mathcal{M}^R} \mathbb{E}\left[d(\mathrm{Exp}(b^R, \boldsymbol{V}^R t_g), C_g^R)^2\right] \tag{3.13}$$

where C_g^R is the covariance matrix subblock defined by features included in R for group g (t_g is our univariate predictor, i.e., time).

To compare the group trajectories, we first parallel transport each tangent vector to the identity as described in §3.2 and then compute the statistic in (3.7) given as $\|\Gamma_{b_1^R \to I} \boldsymbol{V}_1^R - \Gamma_{b_2^R \to I} \boldsymbol{V}_2^R\|_I^2$. In the case of the product space construction, we apply the test in (3.8) to the data subset corresponding to the features in region R, with the same correction as in (3.12).

Summary. We now have a region-based statistic for the manifold regression setting that is approximately normally distributed $N(0, 1)$, allowing effective comparison across differently-sized regions.

Size Correction

A final unresolved yet important issue is that we must correct L_R based on the number of edges $E(R)$ in R. This has a direct consequence on detection power. Observe that the normalization for size correction should be determined by the null distribution of L_R, i.e., when there is no slope difference in the trajectories between groups. In order to derive a correction, we need to characterize the behavior of scan statistics within roughly similar regions, $\max_{R \in \mathcal{R}(A)} L_R$, where $\mathcal{R}(A)$ is the collection of region Rs with similar size as $E(R)$,

$$\mathcal{R}(A) = \{R \in \mathcal{R} : A/2 < |E(R)| \leq A\}. \tag{3.14}$$

Clearly, the behavior of $\max_{R \in \mathcal{R}(A)} L_R$ depends on the "complexity" of $\mathcal{R}(A)$. A clear understanding of how similar subgraphs relate to each other leads directly to a correction tied to their relative sizes.

To investigate the complexity of $\mathcal{R}(A)$, we define the following quantities.

Definition 3.15. *The distance between subgraphs R_1 and R_2 can be given as*

$$d(R_1, R_2) = 1 - \frac{|E(R_1) \cap E(R_2)|}{\sqrt{|E(R_1)||E(R_2)|}} \tag{3.15}$$

Definition 3.16. *Let the ϵ-covering number of $\mathcal{R}(A)$, denoted by $N(A, \epsilon)$, be the smallest integer such that there is a subset $\mathcal{R}_{approx}(A, \epsilon)$ of $\mathcal{R}$ such that*

$$\sup_{R_1 \in \mathcal{R}(A)} \inf_{R_2 \in \mathcal{R}_{approx}(A,\epsilon)} d(R_1, R_2) \leq \epsilon \tag{3.16}$$

where $|\mathcal{R}_{approx}(A, \epsilon)| = N(A, \epsilon)$.

We can verify that all regions in $\mathcal{R}(A)$ can be approximated by regions in $\mathcal{R}_{approx}(A)$ with reasonably small error. From the definitions, notice that the complexity of $\mathcal{R}(A)$ is reflected by $N(A, \epsilon)$. If $N(A, \epsilon)$ is nicely bounded (as is the case here), scan statistics can be calculated very efficiently (Lemma 3.18).

Before stating this result, we make a mild assumption on our graph. For any ball subgraph, the edges around its center are not too sparse, compared to the edges in the outer region of the ball subgraph, i.e., hard on the inside, soft on the outside. This yields,

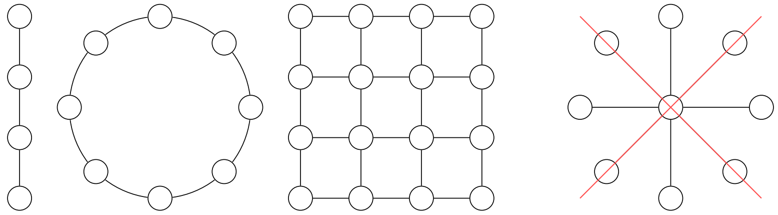


Figure 3.3: (Left) Chain, ring and 2D lattice graphs that satisfy the Avocado Assumption. (Right) Star graph that does not satisfy the property: from the center node the graph is “too dense on the outside."

Assumption 3.17. (*Avocado*) *There exist constants S and H such that, for any $r/2 \leq r' \leq r$ and $v \in \mathcal{V}$,*

$$\frac{|E(B(v,r'))|}{|E(B(v,r))|} \geq H\left(1 - \frac{|E(B(v,r-r'))|}{|E(B(v,r))|}\right)^S. \tag{3.17}$$

We see that this assumption holds for many classes of graphs: a ring graph satisfies this condition when $H = 1$ and $S = 1$ and the 2-D lattice satisfies this condition when $H = 1/4$ and $S = 2$ (see Fig. 3.3). With this assumption, we have the following result for the ϵ-covering number $N(A, \epsilon)$.

Lemma 3.18. *Let $|E|$ be the total number of edges in G. If (3.17) holds and A is given, then, for a constant $C_{H,S}$ which only depends on H and S in (3.17),*

$$N(A,\epsilon) \leq C_{H,S}\frac{|E|}{A}\left(\frac{1}{\epsilon}\right)^{S+1}. \tag{3.18}$$

The proof of this result follows from our ball-subgraph construction and our Avocado assumption and is provided in Appendix A.1.

Intuitively, this result upper bounds the number of graphs that are necessary to search over to completely exhaust the search space of subgraphs. With this result, we can now construct a suitable size correction. Following the work of Walther (2010) and Wang et al. (2016), we can increase the power of our test by using the following statistic:

$$T^* = \max_{R\in\mathcal{R}}\left(L_R - 2\sqrt{\log\frac{|E|}{|E(R)|}}\right). \tag{3.19}$$

The significance of this size correction is that we now have *a single critical value for each* *candidate subgraph, regardless of the subgraph size.* Our final test is defined as $\mathbb{I}[T^* > q_\alpha]$, where q_α is the α-level quantile of T^* under the null hypothesis (that no region is truly significant across groups). By construction, we can control the type 1 error at a specified α-level.

Under the alternative hypothesis of this framework, it is important to note that in many cases, large subgraphs that subsume smaller significant graphs may also have large test statistics, and our hypothesis test only indicates the existence of *some* significant region. To identify or localize the smaller subsets, we follow the procedure from Jeng et al. (2010), by beginning with the subgraph with the largest test statistic and iteratively removing overlapping subsets from the total set of subgraphs. This requires testing each regional/local statistic, $(L_R - 2\sqrt{\log(|E|/|E(R)|)})$ against q_α. Under this procedure, we can control the weak family-wise error rate (wFWER) if we view our problem via the lens of multiple testing. The weak FWER is the probability of false discovery under the null hypothesis. To see that this is inherently controlled, note

$$\mathbb{P}(FN \geq 1|H_0) = \mathbb{P}(T^* > q_\alpha|H_0) \leq \alpha, \tag{3.20}$$

where FN is the number of false discoveries under the null hypothesis. With this correction at the group difference level, we completely avoid any multiple comparisons issues that would arise in the case of a test for each subgraph. In addition to controlling the false positive rate, we have the following guarantee on *identifying truly significant regions* under the normal noise assumption.

Theorem 3.19. *If* (3.17) *holds and the number of edges in the candidate subgraph is larger than* $\log^2|E|$*, i.e.,*

$$|E(R)| \gg \log^2|E| \qquad \forall\, R \in \mathcal{R}, \tag{3.21}$$

then the critical value q_α *satisfies*

$$q_\alpha = O(1). \tag{3.22}$$

Moreover, as $|E| \to \infty$*, if a subgraph* R_0 *obeys*

$$\frac{(\boldsymbol{\beta}_1^{R_0} - \boldsymbol{\beta}_2^{R_0})^T \Sigma_{R_0}^{-1} (\boldsymbol{\beta}_1^{R_0} - \boldsymbol{\beta}_2^{R_0})}{\sqrt{|E(R_0)|}} \gg 2\sqrt{\log\frac{|E|}{|E(R_0)|}}, \tag{3.23}$$

then as $|E| \to \infty$,

$$\mathbb{P}\left(L_{R_0} - 2\sqrt{\log \frac{|E|}{|E(R_0)|}} > q_\alpha\right) \to 1. \tag{3.24}$$

The full proof of this result follows a generic chaining argument (see, e.g. Talagrand (2006)) along with application of concentration inequalities and union bounds, and can be found in Appendix A.1.

Summary. At a high level, this result directly characterizes the behavior of T^* under the null hypothesis H_0 and the alternative hypothesis H_1, respectively. We see that (3.22) implies that T^* can roughly be seen as a constant under the null hypothesis, and under the alternative hypothesis when (3.23) is satisfied, the test based on T^* is consistent, see (3.24).

Workflow for conducting hypothesis tests on temporal trends of graphs

With these guarantees, our full workflow is as follows. First, we use an oracle procedure to generate a graph over our features that roughly captures the conditional independences. Any procedure that provides a conditional independence graph is sufficient. Next, for each ball subgraph over this graph, we compute the Longitudinal-Covariance GLM over these features for both groups, and compute the statistics outlined in Section 3.3. We then compute the size-corrected statistic, and compare against the single critical value. For all regions that pass this threshold, we apply the procedure from Jeng et al. (2010). This workflow shows how to conduct hypothesis tests on temporal trends of large covariance matrices, with improved power and bounded Type 1 error. Additional implementation details can be found in the Appendix A.2.

3.5 Localization Evaluation: Trends of Tobacco Usage Across Gender

We begin our empirical analysis of the model by first applying the subgraph localization procedure by itself (standalone), separate from our manifold regression scheme.

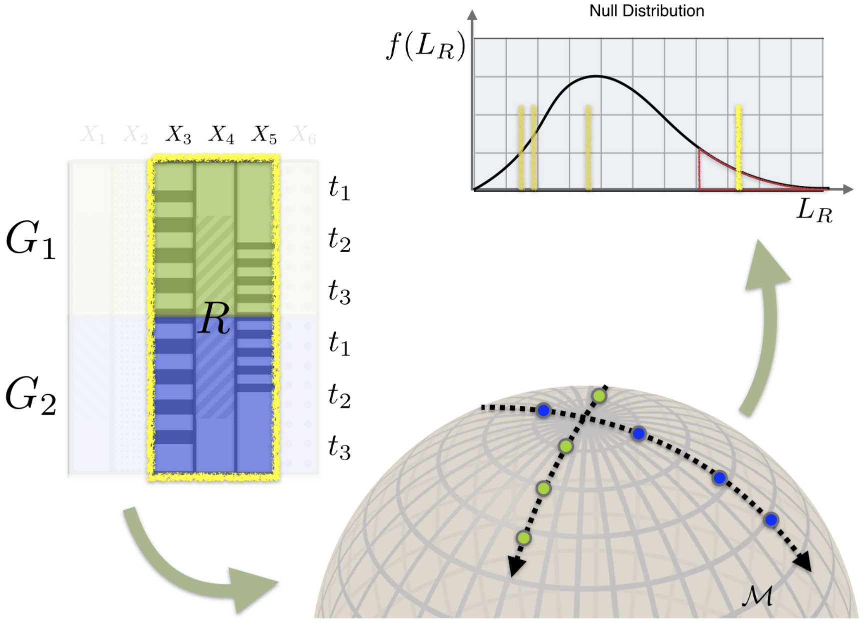


Figure 3.4: The covariance trajectory pipeline. A group of features is selected using scan statistics, manifold trajectories are fit to both groups, and a statistic is computed and tested.

In this case, our statistic is derived from *only* Generalized Linear Models (GLM) constructions, where the $\hat{\beta}_g^R$ in equation (3.12) is the slope estimated from fitting standard first order linear models. Identifying the differentially varying subgraphs across groups in this way is similar to a simpler version of the planted clique identification problem (Arora and Barak, 2009), where the clique we are trying to identify corresponds to those nodes whose slopes vary significantly across groups.

Data. The Center for Disease Control (CDC) provides extensive statistics regarding tobacco and alcohol usage across the US. This data has been collected systematically for the last few decades and is publicly available (includes demographic information and gender). As a simple application of our proposed framework, we may pose the following question: which "sub-groups" of states tend to evolve differently in their correlation (pertaining to tobacco/alcohol usage) over time? Our framework extends easily to answer this question. In this setup, the oracle graph is simply the adjacency graph of the continental US which will be used directly in our scanning procedure. For this dataset, we have direct observations of node measures: the percentage of males and females who reported smoking or drinking heavily in each state. Using

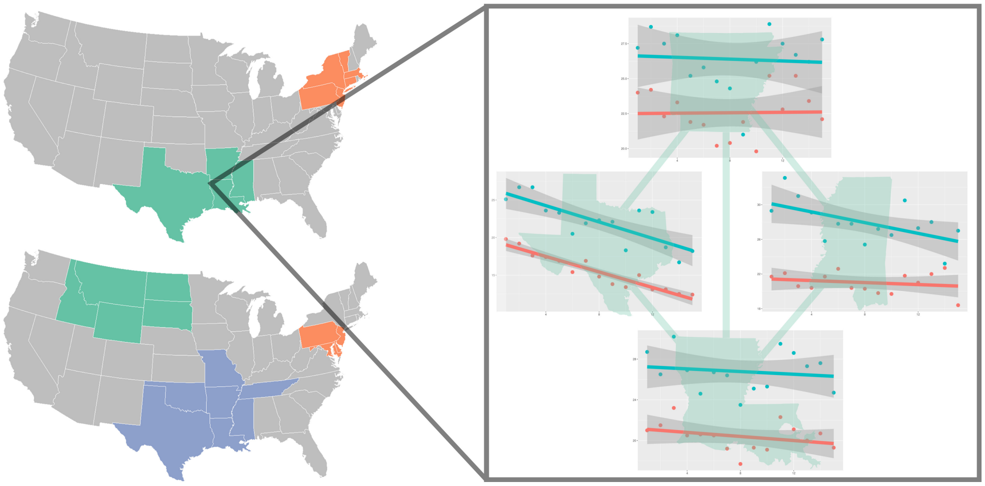


Figure 3.5: (left,top) States identified as having significantly different time-varying tobacco usage across gender from 2001 to 2015. (left,bottom) States identified as having significantly different time-varying heavy drinking use across gender from 2010 to 2015. (right) Linear regressions over tobacco usage fitted to the four states defined by the ball subgraph centered at Louisiana. Best viewed in color.

gender as the group, we fit standard linear models for each candidate subgraph, and compute the difference of gender-wise slopes statistic as described above. In Figure 3.5, we see the regions identified using our method, and interpret some of the tobacco usage findings here.

In the northeast, we see that women have reduced their tobacco usage at a significantly faster rate than men compared to the rest of the country. We suspect that this may be at least partly tied to the development of women's cigarette brands in the late 1960s and 1970s followed by subsequent aggressive public policy campaigns in the 1990s and 2000s to highlight health risks beyond pulmonary or cardiovascular diseases for women (e.g., infertility, reduced bone-density in post-menopausal women). We also see that state-wide indoor smoking bans were put in place in the Northeast ahead of many other states in the union. In the South, the trends among men and women also seems to differ significantly. (see Figure 3.5). Apart from health factors, the group-wise differences in the group-wise trends may also be explained by a few reasons identified in a study in 2007 (Stehr, 2007) which found that as the state sales tax on cigarettes changed (increased), women were significantly more price elastic than men. Between 2006 and 2008, the cigarette tax increased dramatically for all of the 4 states identified *except for Louisiana*, whose tax rate has remained constant. Additionally, while Arkansas did increase their cigarette tax in 2009, they did *not increase taxes in locations near borders shared with higher taxing states*. These intricate relationships among states lend credibility to the fact that our scan statistics framework is indeed identifying interesting sub-regions, and suggests that the full covariance-trajectory pipeline may be more appropriate if effects beyond the means are relevant within an analysis.

3.6 Pipeline Evaluation on Simulations

We next evaluate the ability of our entire analysis pipeline to identify group differences across temporally evolving *covariance* trajectories. In many existing analyses, the effect of the mean differences may be stronger than the effect of the interaction matrix. However, in cases where the *mean signal is weak*, we expect that the covariance effect will be important. To evaluate our model in this regime, we perform a set of simulation studies and also analyze a publicly available longitudinal dataset.

Simulations. We randomly generate SPD matrices from a 'path' of 4 discrete points along the manifold, and use these data as population covariance matrices to generate 0-mean sample data. Table 3.1 shows the results of the hypothesis testing procedure with 50 features averaged over 100 runs, where both the true number of features with covariance trajectory differences, p_t, and the number of samples per group, n, were varied. As expected, our recovery rate increases nicely as a function of the number of samples n and decreases as the size of region of change p_t is increased when n is held constant.

We compare our model to baseline methods that may be used in practice for the foregoing group difference hypothesis test. In standard applications, general linear models (GLMs) are often the first line of attack. When the covariates are assumed to be independent, a simple linear model as in (3.5) may be suitable. However, when the group difference is influenced by specific interactions between covariates, such linear models require additional care. A typical solution is to introduce pairwise interaction terms into the model – a choice between all possible interactions or *specific interactions specified by an expert*. The first model has problems since the number of samples $n \ll p^2$. In the second model, we depend completely on the user's choice of interactions, and must correct for multiple testing when testing different models,

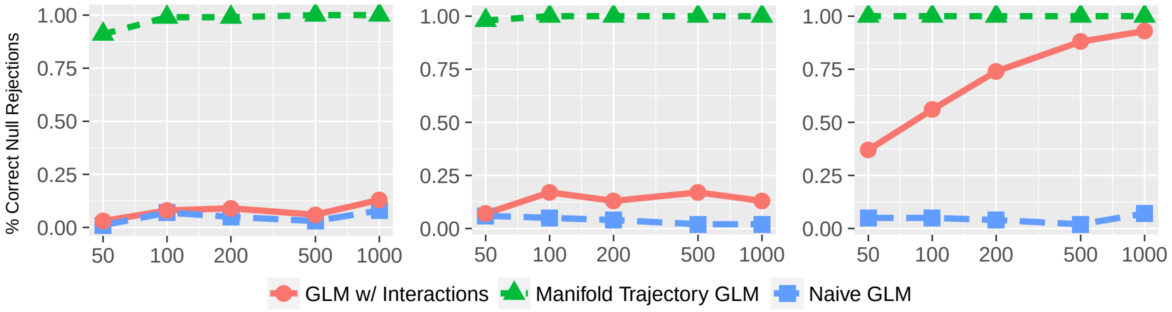


Figure 3.6: Correct null hypothesis rejections over 100 runs for three models. For $p = 50$ features, each plot shows the rejection rate for $p_t \in \{4, 8, 20\}$ (from left to right) respectively as a function of the number of sample points.

Table 3.1: Detection Accuracy of hypothesis test scheme (100 runs).

	$p_t = 5$	$p_t = 8$	$p_t = 10$	$p_t = 15$
$n = 10$	0.06	0.02	0.04	0.03
$n = 20$	0.75	0.75	0.53	0.29
$n = 50$	0.99	1.00	1.00	0.80
$n = 100$	1.00	1.00	1.00	0.95
$n = 200$	1.00	1.00	1.00	0.98
$n = 1000$	1.00	1.00	1.00	1.00

at least partly reducing the power of the final test. Figure 3.6 shows the value of our method over these models. For the interaction GLM case, we randomly select interaction terms to include in the GLM, with size p_t (the ground truth number of variables in the interaction). In this way, we approximate the effect of an oracle specifying to the GLM which terms may describe the underlying interaction. We report the fraction of significance tests where a significance threshold of $p \leq 0.05$ was found for each model, averaged over 100 runs. We see that our proposed scheme consistently achieves near-perfect results in terms of the percentage of null hypotheses that were correctly rejected (i.e., there was a significant group-difference signal). The power of scan statistics on graphs is particularly evident in the needle in haystack setting where the true differential signal is small $(p_t \leq 8)$ and the sample size is small to medium. When the sample size is large and p_t is also large, the standard linear model with additional interaction terms starts to approach the statistical performance of our algorithm.

3.7 Baby Name Trends Over Time

In addition to the simulations above, we report results from a simple analysis of how male/female baby names evolve over time over the last century. The United States Social Security Administration provides a publicly available dataset listing the frequency of the top 1000 baby names in each state for the last 106 years. We evaluate our model in this context to examine which “sub-group” of states tend to evolve (or change) in their “name agreement" (or correlation) over time between boy names and girl names. Here, rather than calculating a sample covariance at each timepoint, we calculate a rank correlation matrix instead. For example, if two neighboring Gulf Coast states, say Georgia and Alabama, substantially agreed on both boys and girls names in the period following the second World War, but gradually this agreement declined over time for girls (but not boys), we expect that our scan statistics on graphs hypothesis test will segment out this differential signal (in slope trends) from the planar graph induced by the states sharing a border. Shown in Figure 3.7 are the regions identified using our method, applied on only the rank correlations for the top 10 names for both genders per state per year. Each highlighted region indicates a sub-group where their “trends of correlation (or agreement/disagreement)” in preferred baby names over the last century varies between boys and girls. For states not identified by our model (in gray), we can conclude that the state-to-state name preference-interactions may have still evolved over time but we have insufficient statistical evidence to conclude that such trends (slopes) are different between boys and girls.

3.8 Identifying Differentially Covarying Features in Preclinical Alzheimer’s Disease

We now describe experiments and results focused on the key motivation of this work — to facilitate analysis of a longitudinal study of individuals at risk for Alzheimer’s disease (AD) where the statistical signal is weak (with small to medium sample sizes). We describe the dataset details followed by the analysis and then interpret

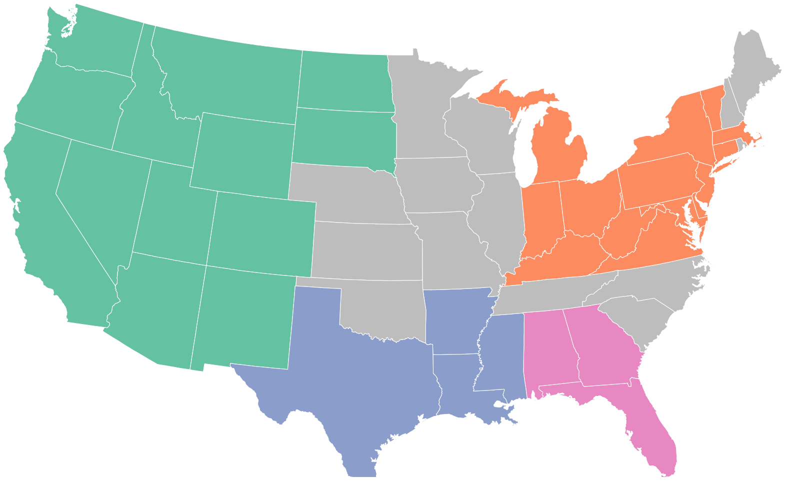


Figure 3.7: Contiguous states identified as having significantly different time-varying co-occurrences between boys and girls baby names from 1910 to 2015. Best viewed in color.

our conclusions in the context of scientific results that have been published in the literature in aging and dementia.

Study background. We analyzed data from a cohort of individuals who have been longitudinally tracked for at least three visits over multiple years, as part of an ongoing study (since 2001) to understand the disease processes in the brain *before* an individual exhibits signs of cognitive decline due to Alzheimer's Disease (AD) (Sager et al., 2005). The study, Wisconsin Registry for Alzheimer's Prevention (WRAP) is among the largest of its kind in existence, focused on "preclinical" AD, i.e., when the individuals are still cognitively healthy, offering a window into the early disease processes where treatments, drugs and interventions are likely to be most effective. WRAP and its ancillary studies acquire neuroimaging data (MRI, PET with different tracers, diffusion MRI) and various clinical test scores, genetic and demographic data as well as clinical measures such as Cerebrospinal Fluid (CSF). Our analysis seeks to understand subtle group-wise differences in longitudinal patterns of dependencies between these measures at this early stage of the disease.

Dataset. The dataset consisted of 114 subjects with imaging data from at least two types of imaging modalities: Positron emission tomography and diffusion weighted Magnetic Resonance (MR) images. Positron emission tomography (PET) images were used to calculate, using well-validated pre-processing pipelines, the mean amyloid-plaque load (an important biomarker for AD) in 16 different anatomical regions of interest in the brain. Amyloid plaque is known to be an AD-related pathology and generally *precedes* onset of cognitive symptoms. Separately, diffusion tensor MR imaging (DTI) data were processed and used to calculate both Fractional Anisotropy (FA) and Mean Diffusivity (MD) in 48 distinct regions (Mori et al., 2008). DTI images provide information about structural connectivity between gray matter regions in the brain. In addition to these 108 $(48 \times 2 + 16)$ image-derived features, we also included in the analysis the participant's scores on a battery of cognitive tests, known to be correlated with various neuropsychological functions (Lezak, 2004). Differences were evaluated on various groupings of the subjects which were, for the most part, based on known results in the literature. Specifically, gender, APOE (Apolipoprotein E) genotype and amyloid positivity (based on thresholding the amyloid plaque summaries) have all been evaluated as significant in AD studies (Racine et al., 2014) but often such analyses involve a population covering a broader disease spectrum where the signal is much stronger.

Is analysis of second order statistics necessary? In Figure 3.8, we present histograms detailing the distribution of two critical cognitive tests, stratified across various groups of scientific interest. Evaluating these distributions were the key motivation for our exploration into the methods described in this chapter. Small differences in means across groups *regardless of grouping selection* (*i.e., stratification variable*), and the saturation that occurs at the ceiling of cognitive test scores and other preliminary experiments conducted by us suggest that standard analyses are not sensitive enough to identify subtle higher-order differences.

Results for Group difference analysis for individuals with imaging data

We now describe, one by one, the components of the largest feature subset discovered for each stratification scheme and highlight the main scientific findings. In most cases, we provide a brief scientific interpretation of the results for the interested reader. Additional details and results can be found in Appendix A.3.

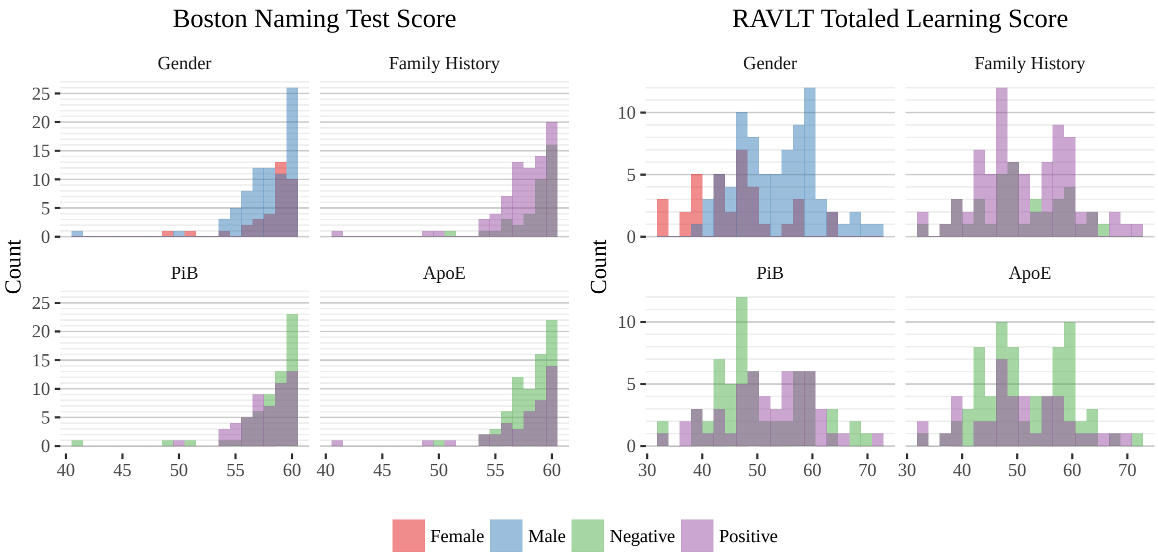


Figure 3.8: Histograms of the Boston Naming Test Scores and RAVLT Total Scores for all time points for the 114 individual measurements across different group separations. The means for each test score is not significantly different across different stratification variable.

A) Graph Scan Statistics on slope differences across gender. The most significant (based on region-score) subset identified by the gender grouping was between the FA DTI measurement in the left cingulum gyrus as well as the scores on the Rey Auditory Verbal Learning Test (RAVLT). In recent AD research, gender has been identified as a factor in the progression of various pathology measures (e.g., incidence and prevalence of AD is higher in women (Fratiglioni et al., 1991; Rimol et al., 2010)), and has contributed to a formal NIH notice (NOT-OD-15-102). However, we note that previous work in the field has *not* identified gender-related differences when looking *only* at diffusion measures in the cingulum (Lin et al., 2014). Our algorithm successfully identified longitudinal changes in *interaction* between these variables which supports the earlier results, and provides some evidence that as men and women age, their cognitive decline as measured by RAVLT manifests differently in relation to the cingulum gyrus.

B) Graph Scan Statistics on slope differences across genotype. Next, we stratified the cohort based on the genotype known to be most closely linked with AD, i.e., the APOE (Apolipoprotein E) gene (Corder et al., 1993) — we inherit one APOE allele from each parent; having one or two copies of the e4 allele increases a person's risk of

Gender	
Set 1	RAVLT Total (1-5)
	FA Cingulum L
Set 2	FA Medial lemniscus L
	FA Cingulum (hippocampus) L
	FA Post thalamic radiation L
Set 3	FA Corticospinal tract R
	FA Superior O.F. fasciculus R

Genotype: APOE4	
Digit Span Backward Raw Score	Stroop Color-word Score
PiB Cingulum Post L	PiB Cingulum Post R
PiB Frontal Med Orb L	PiB Frontal Med Orb R
PiB Precuneus L	PiB Precuneus R
PiB SupraMarginal	PiB Temporal Mid R

Table 3.2: Group difference across Gender (left) and Genotype APOE4 expression (right). Three disjoint sets of features were identified as coavarying significantly differently among gender, while one larger set was identified in the genotype stratification.

getting AD whereas the rarer e2 allele is associated with a lower risk of AD. Using this stratification, we obtain a low-risk and an at-risk group of individuals. Here, we identified amyloid-load regions within the medial and lateral parietal lobes and find that in the “low-risk" group, the covariances between Digit Span and Stroop Color-Word scores (attention and concentration scores) and amyloid load moves from strongly negative towards 0 as a function of age (Table 3.2). In the “at-risk” group (APOE4), however, we find that as a function of age, the features become more and more positively correlated. Existing studies have shown that the accumulation of amyloid is significantly different across APOE4 gene expression (Mormino et al., 2014), and our results provide some evidence that the expression of the genotype may interact with cognitive scores as well, *even at this early stage of the disease*, when the individuals in our cohort are cognitively healthy. The sets of features showing a differential signal are presented in Table 3.2.

C) Graph Scan Statistics on slope differences across amyloid load positivity. As briefly described above, amyloid load is an important biomarker for AD. For our

Amyloid Load (PiB Positivity)		
Set 1	PiB Angular L/R	PiB Cingulum Ant L/R
	PiB Cingulum Post L/R	PiB Frontal Med Orb L/R
	PiB Precuneus L/R	PiB Temporal Sup L/R
	PiB Temporal Mid L/R	**PiB SupraMarginal L**
Set 2	FA Cerebral peduncle R	FA Cerebral peduncle L
	MD Corticospinal tract R	MD Corticospinal tract L
	Trail-Making Test Part A Score	MD Cerebral peduncle R
	PET Cingulum Post R	

Table 3.3: Group difference across Amyloid Load (PiB Positivity)

analysis, amyloid (or PiB) positivity is calculated using the mean amyloid PiB measures across all brain regions using a PiB PET image scan of the participant. When we used this measure for stratification (threshold was set at 1.18, following Darst et al. (2017)), our model identified fifteen of the sixteen PiB regions that were input to the model when the density of the oracle graph was set to be high. This result is as expected, but interestingly we find that controlling for the linear combination of the features (through centering), the residual error *still* has significant signal with the PiB positivity measure, indicating that amyloid burden *interactions* across brain regions plays a very important role in AD progression (Hardy and Selkoe, 2002; Hardy and Higgins, 1992; Tanzi and Bertram, 2005; Jack Jr et al., 2010). When the sparsity of the oracle graph was increased, however, four neighboring regions, the left and right corticospinal tract and the left and right cerebral peduncle were identified on both PiB and DTI measures (supported by the literature (Douaud et al., 2011)), together with Part A of the Trail Making Test (see Table 3.3) which happens to be used in AD diagnosis (Albert et al., 2011). This suggests that changes in atrophy within these regions, as measured by DTI, co-occur with changes in amyloid burden. Additionally, because these regions are highly correlated with rough and fine motor ability (Naidich et al., 2009), it seems plausible that amyloid positivity will lead to higher ‘covariation’ in the regions associated with a measure of fine motor speed, i.e., the Trail Making Test.

Expert Consensus Diagnosis	
WAIS-3 LNS Raw Score	Boston Naming Test Total Score
RAVLT A2 Raw Score	RAVLT A3 Raw Score
RAVLT A4 Raw Score	RAVLT A5 Raw Score
RAVLT A6 Raw Score	RAVLT Delayed Recall Raw Score
Trail-Making Test Part A	Trail-Making Test Part B
Clock Drawing Test Score	CES Depression Scale Score

Table 3.4: Group difference localization across expert clinical diagnosis. With significantly more samples and a larger set of cognitive tests, those above were identified as significantly different across the expert consensus measure.

Results for for Group difference analysis for individuals with Cognitive Testing data

In addition to the dataset presented above, we apply our method to a much larger dataset consisting of approximately 1500 individuals with only cognitive testing data collected in a longitudinal manner. Each individual was administered these tests for between two and three time-points, yielding approximately $n = 4000$ samples for our model. For each assessment, a conference of experts applied a diagnostic label indicating normal cognition or mild cognitive impairment. Using this binary classification, we can stratify our population for group difference analysis. We find that among many different significant subsets, the covariance trajectory among the scores on both parts of the Trail-Making Test and on all trials of the RAVLT test explain a significant group difference. These have previously been shown to be the *most sensitive tests* for early cognitive decline (Albert et al., 2001). Table 3.4 displays the other tests identified by our algorithm, and additional experiments on this larger cohort can be found in Appendix A.3.

Baseline. In various experiments on this dataset, when the MMGLM procedure is performed for the entire feature set in totality (*not* utilizing any of the proposed ideas based on scan statistics), and the null distribution derived using permutation testing, the procedure *yields no significance across any scientifically interesting group stratifications*. This implies that the ability to search over different blocks of the covariance matrix is critical in identifying meaningful group differences in the trajectories, unavailable using alternate schemes. For instance, simpler strategies work well enough for datasets such as ADNI – which includes diseased subjects as well as controls – where the signal is stronger and even temporal modeling may be unnecessary. While the scientific results need to be interpreted with caution and reproducibility experiments on other similar datasets (both within the US and internationally) are in the planning phase, we believe that the ability to localize differences in these interaction patterns in a statistically rigorous manner is valuable and these findings can be investigated standalone, via more classical schemes (e.g., structural equation modeling).

3.9 Conclusion

The analysis of datasets to identify where clinically disparate groups differ is pervasive in biology, neuroscience, genomics and epidemiological studies. We find that graphical models are an ideal tool to analyze high-dimensional data in these areas but have been sparingly used for the analysis of group-wise differences, especially in a longitudinal setting. Motivated by an application related to longitudinal analysis of imaging and clinical/cognitive data from otherwise healthy individuals who are at risk for Alzheimer's disease (AD), we show how a combination of manifold regression with a generalization of scan statistics to the graph setting yields tools that can be directly deployed. We present an efficient algorithm and develop the theoretical results showing the regimes where its application is appropriate. In various experiments, while the standard schemes are not sufficiently powered to detect the signal, our proposed formulation is able to detect meaningful group difference patterns, many of which have a clear scientific interpretation. We believe that these results are promising for the neuroimaging application described and other regimes where group-wise analysis is desired but the number of features is large. We will revisit work in this chapter in some follow-up work, described in Chapter 7.

Chapter 4

Enabling Temporal Neural Networks via Geometric Tensor Representations

The statistical methods in the previous chapter work well with multivariate types of data and scale reasonably well with many features. However, even though we have a linear time scanning procedure, if the number of features reaches thousands or more, computational feasibility can still be an issue. Particularly when dealing with images, using the explicit pixel or voxel-level data is intractable. In these cases, deep learning methods have proven to be extremely effective in modeling high-dimensional data. Combined with methods for temporal modeling, some success has been seen in modeling imaging trajectories. However, a separate problem emerges: the **model size** grows very large, sometimes linearly in the length of the temporal sequence of interest. In this chapter, we examine a novel tensor decomposition that reduces this model size via a **parameter subspace selection**, enabling the modeling of high-dimensional temporal medical imaging data with reasonable computational resources. Work in this chapter originally appeared at the International Conference on Computer Vision (Mehta et al., 2019a).

4.1 Introduction

Recurrent Neural networks (RNNs) and its variants were the de facto tool of choice for modeling sequential data in machine learning and vision. However, these models have been limited in their ability to model high-dimensional data. Part of the reason is that recurrent structures often lead to large model sizes dependent on sequence length, and thus also require an equivalent amount of increased computation. While RNNs have been successfully applied to video data in some cases, the strategy requires problem specific innovations because of the large mapping necessary from inputs to hidden representations. It is fair to say that the growth in the number of model parameters in various types of recurrent models remains a bottleneck for high dimensional datasets. Convolutional neural networks (CNNs), on the other hand, handle high dimensional data far better and can reduce the dimension of an input significantly by deriving rich feature maps. Most computer vision tasks involve some form of a CNN within the architecture, but incorporating CNNs within recurrent structures seamlessly to mitigate the RNN specific model size issues described above is not always straightforward. Notice that a direct replacement of input and output layers with CNNs leads to a shrinkage of the sequence length considerably (Srivastava et al., 2015), and pre-training CNN layers may lead to poor local minima when we train without using an end-to-end pipeline (Donahue et al., 2015). Some recent works suggest the use of dilated convolutional networks for sequence modeling (Yu and Koltun, 2015) to partly mitigate these issues, but this line of work is still developing (Zhen et al., 2019). For model-size reduction, both for RNN style networks and otherwise, PCA or random projections (Ye et al., 2005; Bingham and Mannila, 2001) style “compression" ideas have also been used with varying degrees of success.

Tensor methods provide an interesting perspective on the effective degrees of freedom afforded by a given network, acting as a surrogate for the actual “size” of the architecture. Decomposition-based methods have been shown to enable low dimensional representations of very high dimensional data (Hwang et al., 2018; Novikov et al., 2017), and while these ideas were known to be effective in the “shallow" regime much earlier, new results also demonstrate their applicability for deep neural networks. In particular, a number of tensor based methods have been successfully adapted for deep neural network design and compression (Cohen et al., 2016; Zhang et al., 2017; Yu et al., 2017; Xiong et al., 2019a). Specifically, Yang et al. (2017) shows that these methods can be very effective in reducing the parameter cost of weight layers in RNNs, enabling simple video analysis tasks that previously would have been computationally prohibitive.

There are a number of key reasons why the size of the model, especially in the context of formulations for sequential data, is central here. Our goal is to design rich sequential or recurrent models to analyze a longitudinal sequence of high dimensional 3D brain images. This task raises two issues. **First**, unless the model size is parsimonious, we find that merely instantiating the model with data involving 3D images over multiple time points, even on multiple high end GPU instances, is challenging. **Second**, the eventual goal of medical image analysis is either scientific discovery or generating actionable knowledge for patients. Both goals require evaluating a model's confidence via classical or contemporary statistical techniques: for instance, how confident is the model in its prediction? Most, if not all, available tools for assessing model uncertainty of deep neural network models have a strong dependence on the number of parameters in the model. Therefore, even if the first issue above could be mitigated by clever implementation ideas, purely as a practical matter, the design of rich and expressive models with a small number of parameters yields immense benefits for calculating model uncertainty.

Contributions. We tackle the problem of modeling sequential 3D brain imaging data using recurrent/sequential models. Our development starts from well known results on tensor decomposition. In particular, we make use of the tensor train representation, which has been shown to be effective in several applications in vision and machine learning. We derive a reformulation of the decomposition using orthogonality constraints and show that while this makes the estimation slightly more challenging, it reduces the number of parameters needed significantly. With this **parameter subspace** identified, we present a novel estimation scheme based on Stiefel manifold optimization and demonstrate how the end to end construction yields benefits for convergence and uncertainty estimation. Finally, from the empirical side, we discuss how we enable analysis of and prediction using sequential 3D brain imaging datasets, which to our knowledge is the first such result using deep recurrent architectures.

4.2 Orthogonal Tensor Trains

Recall from Chapter 2 that a tensor X can be efficiently represented and operated on via the tensor train decomposition. As described, a number of tensor train operations with respect to approximation and projection require computing the QR decomposition of matricized cores. In the applications for which tensor trains were originally developed, these operations were necessary (Oseledets, 2011; Klus et al., 2018). For modern neural network applications, where the tensor operator may be our target of *learning*, it may be sufficient to treat each matrix product as its own variable, and through the standard TT decomposition learn the cores along the **product of Stiefel manifolds**.

A naïve approach may orthogonalize the reshaped cores (see (2.27)), and progressively push the upper triangular part of the core decomposition into the next core, resulting in the following exact formulation with appropriate reshaping:

$$\begin{aligned} T &= A_1^L A_2^L \cdots A_d^L \\ &= Q_1^L R_1 A_2^L \cdots A_d^L = Q_1^L \left(R_1 A_2^L \right) \cdots A_d^L \\ &= Q_1^L Q_2^L R_2 \cdots A_d^L \\ &= Q_1^L \cdots Q_d^L R_d \end{aligned} \tag{4.1}$$

where $[Q_1^L, R_1] = qr(A_1^L)$, $[Q_i^L, R_i] = qr(R_{i-1}A_i^L),\ i \in \{2, \ldots, d\}$, and $Q_i^L \in \mathbb{R}^{r_{i-1}n_i \times r_i}$ with $R_i \in \mathbb{R}^{r_i \times r_i}$. Each Q_i^L is a point on the Stiefel manifold given by St$(r_i, r_{i-1}n_i)$. Here, the number of components in the product space of Stiefels is d, with the eventual 'residual' $R_d \in \mathbb{R}$. This decomposition is exact and only requires a reshaping of the tensor cores. If all $r_i = r, n_i = n$, then the total number of parameters needed is

$$\begin{aligned} &\sum_{i=1}^{d} \left[nr^2 - \frac{r(r+1)}{2} \right] + \frac{r(r+1)}{2} \\ &= dnr^2 - d\frac{r(r+1)}{2} + \frac{r(r+1)}{2} \\ &= dnr^2 - (d-1)\frac{r^2+r}{2}, \end{aligned}$$

compared to the full format with dnr^2 total parameters. It is important to note that in this formulation, the TT cores themselves are **not** orthogonal. Reshaping is required to bring the matricized form back to TT-cores of size $r_{i-1} \times r_i$, and in practice it is not easy to perform simple TT-tensor multiplication in this form. Additionally, we now need to optimize over Stiefel manifolds of a larger size, namely $O(nr^2)$.

A Nicer Tensor Train Approximation

Ideally, we would prefer a construction which keeps the standard TT-core format and involves optimization over "smaller" Stiefel manifolds. Consider the following representation, in which each TT-core itself is orthogonal.

Definition 4.1. (*Orthogonal Tensor Train*) *The Orthogonal Tensor Train is defined as*

$$T(x_1, \ldots, x_d) = Q_1(x_1) \cdots Q_d(x_d), \tag{4.2}$$

where each $Q_i(x_i)$ lies on the Stiefel St(m_i, M_i), and $m_i = \min(r_{i-1}, r_i),\ M_i = \max(r_{i-1}, r_i)$.

While in this formulation the total number of components in the product space of Stiefels is nd, the dimension of each manifold is **significantly smaller**, dependent *only* on the core rank as opposed to the mode size. The total number of parameters, if $n_i = n, r_i = r$, is

$$n \sum_{i=1}^{d} \left[r^2 - \frac{r^2 + r}{2} \right] = dnr^2 - dn\frac{r^2 + r}{2}. \tag{4.3}$$

When compared to the full TT representation, the Orthogonal Tensor Decomposition (OTT) requires $(r+1)/2r \approx 1/2$ as many parameters. If $r_i = r_{i+1}$, then St$(m_i, M_i) = SO(m_i)$, where SO is the special orthogonal group.

This construction can be seen as an approximation to the full tensor train format, in which the upper triangular part of each core is set to identity:

$$\begin{aligned} T(x_1, \ldots, x_d) &= A_1(x_1) \cdots A_d(x_d) \\ &= Q_1(x_1) R_1(x_1) \cdots Q_d(x_d) R_d(X_d) \\ &\approx Q_1(x_1) \cdots Q_d(x_d) \end{aligned} \tag{4.4}$$

Is this useful? It is not obvious that this construction is useful at all. How much is lost through this approximation? What is gained by using this construction? In what follows, we demonstrate that we can approximate any tensor with bounded norm using an OTT, and that with a full rank assumption and a trainable constant, our formulation admits a solution with ϵ error.

Theoretical Analysis

We start by reshaping any tensor T to a matrix T^M by grouping the modes into two groups, $T^M \in \mathbb{R}^{n \times m}$. We may fix this arbitrary matrix as $T^M = A \in \mathbb{R}^{n \times m}$.

Proposition 4.2. *Given a 2D tensor $A \in \mathbb{R}^{m\times m}$, $A_{ij} \in [-1, 1]$, there exist sets of unit vectors, $\{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^m$, $\{\mathbf{y}_j\}_{j=1}^m \subset \mathbb{R}^m$ such that, $\forall \epsilon > 0$, $\|A - \widetilde{A}\| < \epsilon$, where, $\forall i, j,\ \widetilde{A}_{ij} = \mathbf{x}_i^t \mathbf{y}_j$.*

Proof. Let $A = USV^T$ be the SVD of A. Let $\epsilon > 0$, we will perturb S along the diagonal to generate $\widetilde{S}$ such that, $\|S - \widetilde{S}\| < \epsilon$. Let $X = [\mathbf{x}_i]$ and $Y = [\mathbf{y}_i]$. We will first give an algorithm to generate $\widetilde{X}$ and $\widetilde{Y}$ with each of its column being orthonormal such that, $\widetilde{X}^T\widetilde{Y} = S$. Then, $X = \widetilde{X}U^T$ and $Y = \widetilde{Y}V^T$.

We begin with an algorithm for $m = 3$. Choose $\{\widetilde{\mathbf{x}}_i\}$ to be unit vectors and assign $\widetilde{\mathbf{y}}_3 = \widetilde{\mathbf{x}}_1 \times \widetilde{\mathbf{x}}_2$, $\widetilde{\mathbf{y}}_2 = \widetilde{\mathbf{x}}_3 \times \widetilde{\mathbf{x}}_1$. Then, make $\widetilde{\mathbf{y}}_2$ and $\widetilde{\mathbf{y}}_3$ to be of unit length. Now, rotate $\widetilde{\mathbf{x}}_2$ in the plane spanned by $\{\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2\}$ such that, $\widetilde{\mathbf{x}}_2^t\widetilde{\mathbf{y}}_2 = \widetilde{S}_{22}$. Similarly, rotate $\widetilde{\mathbf{x}}_3$ in the plane spanned by $\{\widetilde{\mathbf{x}}_3, \widetilde{\mathbf{x}}_1\}$ such that, $\widetilde{\mathbf{x}}_3^t\widetilde{\mathbf{y}}_3 = \widetilde{S}_{33}$. Now, assign, $\widetilde{\mathbf{y}}_1 = \widetilde{\mathbf{x}}_2 \times \widetilde{\mathbf{x}}_3$ and make it unit length. Now, fixing $\widetilde{\mathbf{x}}_2$ and $\widetilde{\mathbf{x}}_3$, the above steps are a continuous mapping, F from $\mathbf{S}^2$ to $[-1, 1]$, i.e., by changing different $\widetilde{\mathbf{x}}_1 \in \mathbf{S}^2$, we will get different values for $\widetilde{\mathbf{x}}_1^t\widetilde{\mathbf{y}}_1$. Also, notice that, if, for a particular choice of $\{\widetilde{\mathbf{x}}_i\}$, $\widetilde{\mathbf{x}}_1^t\widetilde{\mathbf{y}}_1 > 0$, then, for the choice of $\left\{\widetilde{-\mathbf{x}_i}\right\}$, the above construction returns $-\widetilde{\mathbf{y}}_2$ and $-\widetilde{\mathbf{y}}_3$ and F returns, $-\widetilde{\mathbf{x}}_1^t\widetilde{\mathbf{y}}_1 < 0$. Thus if $a \in F\left(\mathbf{S}^2\right)$, $-a \in F\left(\mathbf{S}^2\right)$. Furthermore, $1 \in F\left(\mathbf{S}^2\right)$ and hence, $-1 \in F\left(\mathbf{S}^2\right)$. As $\mathbf{S}^2$ is connected and F is continuous, $F\left(\mathbf{S}^2\right)$ is connected, and so, $\exists\, \{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^m$ and $\{\mathbf{y}_j\}_{j=1}^m \subset \mathbb{R}^m$, s.t., $(\forall\, \{i, j\})$, $\widetilde{\mathbf{x}}_i^t\widetilde{\mathbf{y}}_j = \widetilde{S}_{ij}$. Since $\|S - \widetilde{S}\| < \epsilon$ and the choice of $\epsilon > 0$ is arbitrary, we can see that $\|A - \widetilde{A}\| < \epsilon$.

Using the generalization of cross product by exterior algebra, the above procedure can be naturally extended to arbitrary $m > 3$. □

A direct corollary of the above result allows approximating an arbitrary 2D matrix,

Corollary 4.3. *Given a 2D tensor $A \in \mathbb{R}^{m\times m}$, there exists sets of unit vectors, $\{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^m$, $\{\mathbf{y}_j\}_{j=1}^m \subset \mathbb{R}^m$ and fixed constant c such that, $\forall \epsilon > 0$, $\|A - \widetilde{A}\| < \epsilon$, where, $\forall i, j$, $\widetilde{A}_{ij} = c\mathbf{x}_i^t\mathbf{y}_j$.*

Proof. Given any arbitrary matrix A, define $A' = A/|A|_\infty$. Then $A'_{ij} \in [-1, 1]$, and by Proposition 4.2 we can construct unit vectors $\mathbf{x}_i, \mathbf{y}_j$ such that $\forall \epsilon > 0$, $\|A'_{ij} - \mathbf{x}_i^T\mathbf{y}_j\| < \epsilon$. Then immediately $\forall A_{ij}$, we have $A_{ij} = cA'_{ij}$ where $c = |A|_\infty$. □

We also have the following directly from Proposition 4.2.

Corollary 4.4. *Given a 2D tensor $A \in \mathbb{R}^{m\times m}$, with $\|A\|_F \leq 1$, there exists a set of orthonormal matrices $\{B_i\} \subset SO(m)$ and a set of unit vectors $\{\mathbf{y}_j\}_{i=1}^m \subset \mathbb{R}^m$ such that $\forall \epsilon > 0$, $\|A - \widetilde{A}\| < \epsilon$, where, $\forall i, j,\ \widetilde{A}_{ij} = \mathbf{1}^t B_i^t \mathbf{y}_j$.*

Example 4.5. *Applying the above result to OTT, equivalence is relatively straightforward to show. Consider the problem of approximating a 4 dimensional tensor T with $n_{1,2,3,4} = n = r$. Let $Q_1(x_1) \in \mathbb{R}^{1\times n}, Q_2(x_2), Q_3(x_3) \in \mathbb{R}^{n\times n}$, and $Q_4(x_4) \in \mathbb{R}^{n\times 1}$. By Corollary 4.4 we can write two vectors indexed by x_1, x_2 and x_3, x_4 as $T^A(x_1, x_2) = Q_1(x_1)^\top Q_2(x_2)$ and $T^B(x_3, x_4) = Q_3(x_3)Q_4(x_4)$ respectively. The multiplication of these vectors T^A, T^B again yields a single element indexed by x_1, x_2, x_3, x_4, which can take any value between $[-1, 1]$ by Proposition 4.2. Then clearly the cores Q form an equivalent definition of $\mathcal{X}$.*

We can then apply Corollary 4.4 and find that the product of indexed orthonormal matrices and orthonormal vectors with full rank can approximate any matrix with bounded norm. Applying this to our OTT format, it immediately follows that with the addition of at most dn constants in $\mathbb{R}$ we can approximate any arbitrary tensor. While this addition would put the format well over the number of parameters in the standard format, this provides sufficient evidence that, in typical learning settings in which our model is already overparameterized, we can still capture the full expressive power of the model class in which an OTT format is inserted.

Remark. It also important to note that the above calculation of dimensionality is the *intrinsic* dimension. The number of actual allocated variables is indeed dn^3 for an exact formulation. It remains open to theoretically analyze the degradation of the approximation as $r < n$.

Efficient Stiefel Optimization

Here, we describe how to compute an OTT approximation of a tensor T, which can be posed as the following minimization problem.

$$\begin{aligned} \min_{\{Q_i(x_i)\}_{i=1}^d} E = \sum_{\{x_i\}} \|T(x_1, \ldots, x_d) - Q_1(x_1) \cdots Q_d(x_d)\| \\ \text{s.t.} \quad Q_i(x_i)^\top Q_i(x_i) = I_p \quad \forall i, x_i \end{aligned} \tag{4.5}$$

Notice that this optimization is difficult because of the orthogonality constraint (Edelman et al., 1998; Collins et al., 2014). An efficient way to solve this is by doing the optimization on the product of (compact) Stiefel manifolds: let it be denoted by $\mathcal{P}_S$. We will use the product ℓ_2 metric on this product space. Given $x_1, \cdots x_d$, we perform an optimization on the product of Stiefel manifolds to solve for $\{Q_i(x_i)\}$ for $i \in [1 \ldots d]$.

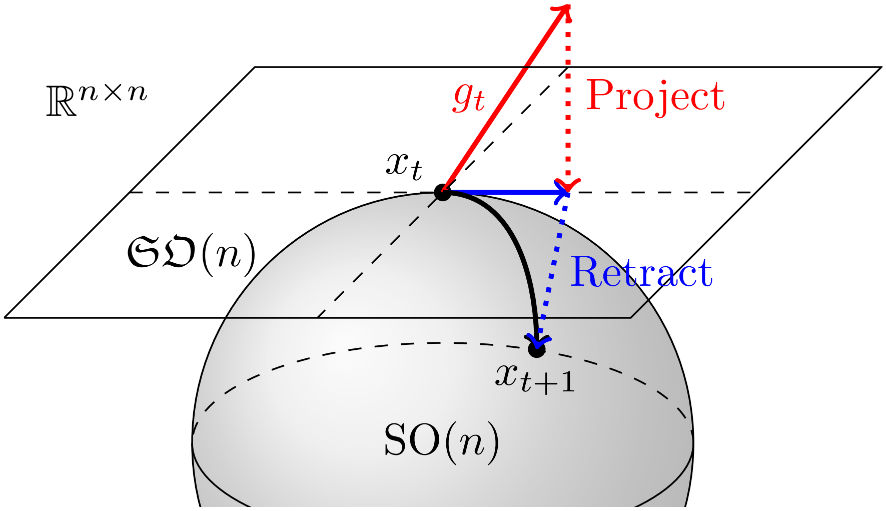


Figure 4.1: Gradient descent algorithm using the projection and retraction on the Stiefel manifold. The update is applied to each core individually, allowing for smaller manifold operations that would otherwise scale poorly with dimension.

We use a Riemannian gradient descent technique on this product of Stiefel manifolds $\mathcal{P}_S$. Given $\{Q_i^t(x_i)\}$ as the solution of the t^{th} step, the $(t+1)^{th}$ solution, $\{Q_i^{t+1}(x_i)\}$, can be computed using

$$\left\{Q_i^{t+1}(x_i)\right\} = \text{Exp}\left(\left\{Q_i^t(x_i)\right\}, \frac{\partial E}{\partial\left\{Q_j^t(x_j)\right\}}\right), \tag{4.6}$$

where Exp is the Riemannian Exponential map on $\mathcal{P}_S$. On $\mathcal{P}_S$, the product of Stiefels, computation of Riemannian Exponential map is not tractable and needs an optimization, in contrast to the SPD manifold discussed in Chapter 3. Hence, we use a Riemannian retraction map as proposed in Kaneko et al. (2013).

Figure 4.1 and Algorithm 2 summarize this procedure. For each orthogonal core, the gradient is computed with respect to the Euclidean ambient space and projected to the tangent space at the current iterate. The update is constructed by moving back to the Stiefel with the Riemannian exponential map.

Square Stiefels/SO(n)

In practice, when learning an OTT operator, we will primarily be setting the rank to be fixed for all cores. The Stiefel manifold, St(n, n) with $n = p$ is equal to the special orthogonal group SO(n). The Riemannian Exponential map on SO(n) is the matrix

Algorithm 2: Stochastic OTT Optimization

```
for $t$=1,...,$T$ do
    $g_t := \frac{df}{d\mathcal{W}} f(X^{mini-batch})$
    for Core $Q_t^i \in \mathcal{W}_t$ and Core Gradient $g_t^i \in g_t$ do
        $G_t^i = P_{T_{\mathcal{W}_t}M}(g_t^i)$            ▷ Projection Step
        $Q_{t+1}^i \leftarrow \exp(Q_t^i, G_t^i)$          ▷ Retraction Step
    end
end
```

exponential, computationally intensive to both compute and backpropagate through. Hence, we use the *Cayley map* from $\mathfrak{SO}(n)$ to SO(n), given by $A \mapsto (I - A)(I + A)^{-1}$, where $\mathfrak{SO}(n)$ (the space of $n \times n$ skew-symmetric matrices) is the tangent space of SO(n) at identity. Although the Cayley map requires a matrix inverse, it is much easier to handle using standard tools in modern toolboxes (e.g., TensorFlow, PyTorch). Observe that the work in Helfrich et al. (2017) used the Cayley map for RNNs, but does not make use of the sparse representation of a skew-symmetric matrix $A \in \mathfrak{SO}(n)$. In contrast, in our formulation we use the Cayley map as a mapping from $\mathbb{R}^{\frac{n(n-1)}{2}}$ to SO(n). This enables a strict reduction in the number of trainable/learnable variables in a network, and provides a direct path through which gradients can be computed and backpropagated. Algorithm 3 describes the procedure for constructing an OTT-core. The Euclidean variable vector w is mapped directly to the upper triangular part, defined as $\mathrm{triu}(\cdot)$, of a new matrix R, and by subtracting its transpose, we arrive at a skew symmetric matrix A. The Cayley map, as described above, maps to our Orthogonal OTT Core.

Remark. Note that the Cayley map is not a bijective mapping between $\mathfrak{SO}(n)$ and SO(n) as the range is not the entire SO(n). This is because the Cayley map cannot generate matrices with negative eigenvalue(s). Empirically, we do not find this to be an issue when learning the OTT representation directly, and hypothesize this may have desirable downstream qualities enabling better performance at training time (positive definiteness is often desirable for fast *convex* optimization).

With this efficient approximation in hand, we are able to directly apply our OTT formulation to architectures for a variety of applications.

Algorithm 3: Constructing an OTT Variable

```
Function OTT-Variable($d, n_{in}, n_{out}, r$):
    $OTT \leftarrow \emptyset$
    for $i \in 1, \ldots, d$ do
        for $j, k \in 1, \ldots, n_{in}[i], 1, \ldots, n_{out}[i]$ do
            $OTT$.append(OTT-Core($r$))
        end
    end
    return $OTT$
Function OTT-Core($r$):
    $w \leftarrow \mathbb{R}^{r(r-1)/2}$
    $R \leftarrow \text{triu}(w)$
    $A \leftarrow R - R^\top$
    $Q \leftarrow (I - A)(I + A)^{-1}$
    return $Q$
```

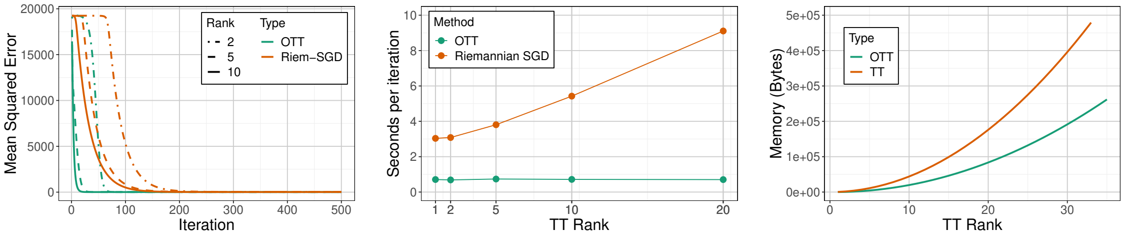


Figure 4.2: (left) Mean squared error for different TT-ranks, using both the Riemannian formulation (2.28) and the approximate Stiefel formulation (4.1). (center) Effect of TT-rank on per iteration runtime of both methods. OTT is significantly faster (10x) than the Riemannian formulation. (right) Memory Dependence of both TT and OTT constructions as a function of rank. The OTT formulation allows for models roughly double the size of TT.

4.3 Evaluating performance on Simulations, Moving MNIST and Video data

First, we evaluate how well our OTT formulation performs relative to existing methods, on synthetic datasets as well as other popular datasets used for sequential deep models. All synthetic experiments were conducted using a Tensorflow implementation on an Nvidia Titan Xp GPU.

(A) OTT vs Riemannian SGD on synthetic data. To empirically verify the claims in Section 4.2 and to evaluate the value of our OTT construction over the existing

Riemannian SGD framework, we simulate a simple least squares problem with the goal of learning a tensorized weight matrix,

$$\min_{W_{TT}} \sum_{i=1}^{n} ||y_i - W_{TT} x_i||^{\cdot} 2$$

Here we use the naïve but exact OTT construction, using the optimization scheme in Section 4.2. A weight matrix W is initialized to a random matrix with size 784×625, and samples are drawn from $y = Wx$. The matrix is reshaped as a tensor with modes $[4, 7, 4, 7] \times [5, 5, 5, 5]$.

Results. Figure 4.2 shows the convergence rates of both methods with fixed learning rates for various TT-ranks. (**a**) **Quality and speed.** For this toy problem, not only is the OTT construction able to find a good solution, it is able to find it significantly faster than Riemannian SGD. (**b**) **Update steps.** Additionally, we note that the time per iteration is significantly shorter for the OTT construction. OTT allows for each manifold update step to be performed on a low dimensional Stiefel, and so retraction and projection is *fast*. The Riemannian method requires left orthogonalization and QR decompositions of larger matrices, leading to a slower, TT-rank dependent runtime, shown in Figure 4.2. (**c**) **Memory footprint.** Finally, we see in Figure 4.2 that the memory consumption of OTT is quite modest compared to TT (which already offers significant memory savings over alternative existing schemes). This may be a beneficial feature when running a large sequential model on less expensive GPUs. Given these results, we use a basic SGD update for TT in subsequent experiments.

(**B**) **Moving MNIST.** The moving MNIST dataset (Srivastava et al., 2015) consists of handwritten digits moving within a specified larger image. We first demonstrate that for simple sequences, reconstruction under a complete tensor train framework is possible, and representing fully connected layers with an OTT layer reduces the number of parameters **without** image degradation. Here, we use a vanilla RNN, with a state size of 4096 and TT-Rank 64.

Results. Figure 4.3 shows the ground truth and reconstruction results for images with size 256×256, where each sequence is of length 8, and the direction and orientation of the digit is random. (**a**) **Reconstruction accuracy and model size.** The

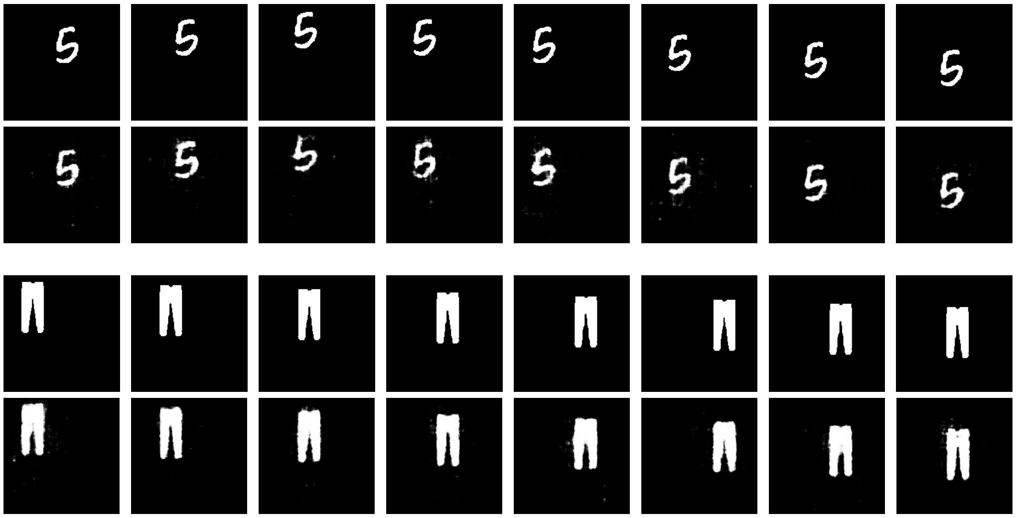


Figure 4.3: Sample ground truth (top) and reconstruction (bottom) of Moving MNIST digit and fashion sequences of size 256×256. We see good consistency between each upper/lower rows for both datasets.

entire recurrent network is compressed with OTT layers for input-to-hidden, hidden-to-hidden, and hidden-to-output maps. With a large state size of 4096, we are able to nicely capture and rebuild the entire sequence with a significantly smaller model size. **(b) Scaling to larger images.** This effective compression also allows us to scale up – to significantly larger images of size 1024×1024, *with no loss in reconstruction quality*, without the need for more sophisticated convolutional architectures.

(C) Hollywood2. We find that these results extend nicely to LSTMs/GRUs and for classification tasks as well. The Hollywood2 dataset (Marszałek et al., 2009) consists of video clips from 69 movies labeled with 12 different actions from "answering the phone" to "driving a car" (Figure 4.4). Following the preprocessing steps of Yang et al. (2017), we feed resized clips of size $234 \times 100 \times 3 \times T$ to our model, where the length of a sequence (number of frames) T ranges from 29 to 1496. We tensorize the input as $10 \times 18 \times 13 \times 30$ for all input sequences (padded to 1496) and the hidden states as $4 \times 4 \times 4 \times 4$, with TT and OTT ranks set as 4.

Results. Tensor trains here allow us to completely operate on the *entire video sequence.* **(a) Parameter size.** The number of parameters in our model is a few thousands (1864 for OTT, 3104 for TT) compared to millions needed for a standard fully connected model. **Accuracy comparison.** Using Mean Average Precision

Figure 4.4: Sample sequences from the Hollywood2 dataset. Labels are (Top) Answer Phone, (Middle) Drive Car, and (Bottom) Get Out Car.

(MAP) as a measure of accuracy for this multi-label problem, we find that using an OTT-LSTM or OTT-GRU in place of a TT-LSTM or TT-GRU leads to *no significant difference in MAP.*

4.4 Identifying Differential Progression in AD

Motivation. The Alzheimer's Disease Neuroimaging Initiative (ADNI, adni.loni.usc.edu) provides a comprehensive dataset targeted towards understanding AD. The goals of the initiative include measuring the development of the disease as a function of different imaging modalities, other biological markers, and clinical and neuropsychological assessments. Deep learning methods traditionally applied to this corpus require imaging data to be heavily preprocessed into summary measures, such as *regions of interest*. In other cases, based on the needs of the application (e.g., segmentation), the approach may operate with 3D image patches instead of the entire image. The size of the images, especially when considered longitudinally, can be impractical for modern deep learning frameworks unless some novel implementation tricks are utilized.

Data. Our dataset consists of 522 subjects with Magnetic Resonance Imaging (MRI) scans collected over three years. For each individual, an MRI was collected annually, along with a battery of neuropsychological evaluations.

Pre-processing. Full head MRIs were processed using SPM12 Ashburner et al. (2014). Each image was segmented/registered using the MNI152 template. Gray matter probabilities were computed, and these gray matter density (GMD) images were used as input to our models. The processed image size was $121 \times 145 \times 121$ (voxel size $1.5mm^3$), with **3 images per subject**.

Model. At this scale, we use convolutional input and deconvolutional output layers to incorporate local information with respect to reconstruction and prediction. The architecture consists of a straightforward 3-state RNN with input-hidden, hidden-hidden, and hidden-output layers replaced with TT and OTT layers. Input volumes are passed through a 3D convolutional input network, with max-pooling layers and ReLUs. Hidden states are passed through an output convolutional network consisting of max-unpooling layers using indices saved from the input CNN. Strides were fixed at 1 with a kernel size of $3 \times 3 \times 3$, with successive convolutions decreasing (increasing) the number of channels by 2. Max pooling was applied uniformly to all 3 input channels with a stride of 2. Adam optimization (Kingma and Ba, 2014) was used for all ADNI experiments, with learning rate $1e^{-3}$, and decay rate 0.9 and 0.999 for the first and second moments. TT and OTT layers were fixed with a rank of 64. Batch sizes were fixed at 4.

Modeling gray matter progression in AD

Our first goal is to predict the next MR image given the previously seen ones. Importantly, standard RNN constructions cannot easily handle inputs of this size. On a single NVIDIA Titan Xp, the images must be **downsampled** by over 20× to allow for a batch size of 4 in a standard LSTM model with a hidden state size of 2048 (4.3 billion parameters for a full sized input map).

Results. Fig 4.5 show the results for a held-out subject in the study using OTT-RNN on a single representative 2D slice, with their predicted third timepoint image. While higher levels of compression (lower OTT ranks) lead to "blocky" reconstructions, our model is still able to identify boundaries of edges between low and high probabililty voxels.

Cognition from gray matter sequence

Based on the results from the above experiment, one may ask if, in fact, a good model of progression is being learned, or if only the "average" of all participants is being predicted by the model.

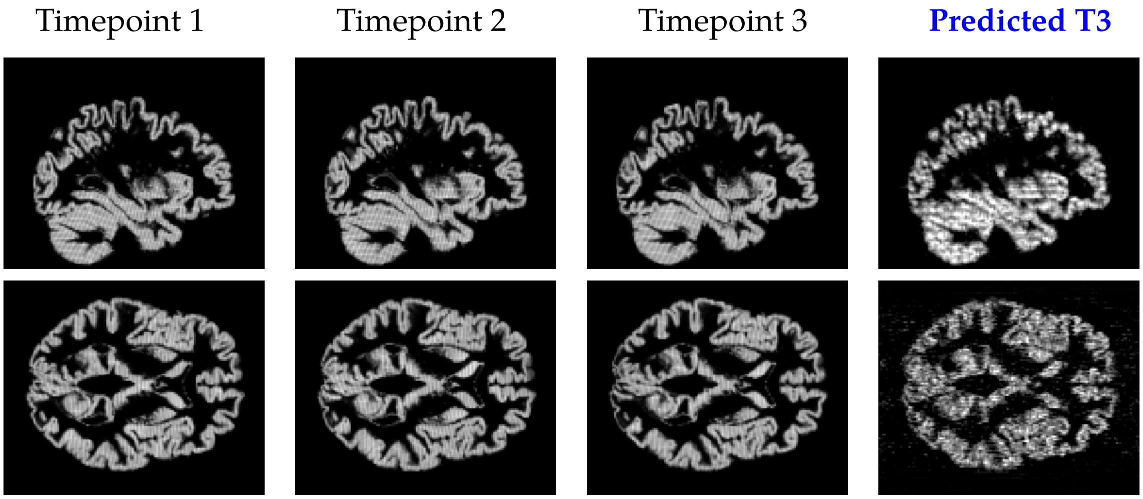


Figure 4.5: Ground truth progression and prediction of gray matter probabilities in an individual from our validation set. From the left, the first three images are the ground truth images at visits 1, 2, and 3, followed by our prediction at visit 3.

(A) Predicting Cognition from 3D image sequences. To answer this question, we can directly try to predict summary cognition measures which are used in practice. Diagnoses themselves can often be based on partial information available to medical experts at that time. Indeed, a small number of individuals in the ADNI cohort have been diagnosed with AD or mild cognitive impairment and have *regressed* to a cognitively healthy diagnosis at their next visit. In these situations, categorical diagnoses can be seen as a noisy summary measure of decline. We predict a real-valued measure collected at each timepoint. As in the previous chapter, The Rey Auditory Verbal Learning Test (RAVLT) evaluates a large variety of cognitive functions, including short and long term memory, cognitive function, and learning ability (Schmidt et al., 1996), and has been identified as a strong indicator for developing AD pathology. We train both TT and OTT models with dropout for 200 epochs.

Results. Figure 4.6 (left) shows the results of this analysis. Here, the advantage of the OTT construction is clear, we are able to converge significantly faster compared to the TT construction, with half as many parameters.

(B) Quantifying Model Uncertainty. Broad application of deep learning models in neuroimaging remains limited, namely due to sensitivity of black box models to mild perturbations in input data or model parameters, leading to unreliable predictions. MC Dropout (Gal and Ghahramani, 2016), approximates model (epistemic) uncertainty by using dropout *at prediction time*. Simulating an ensemble of networks

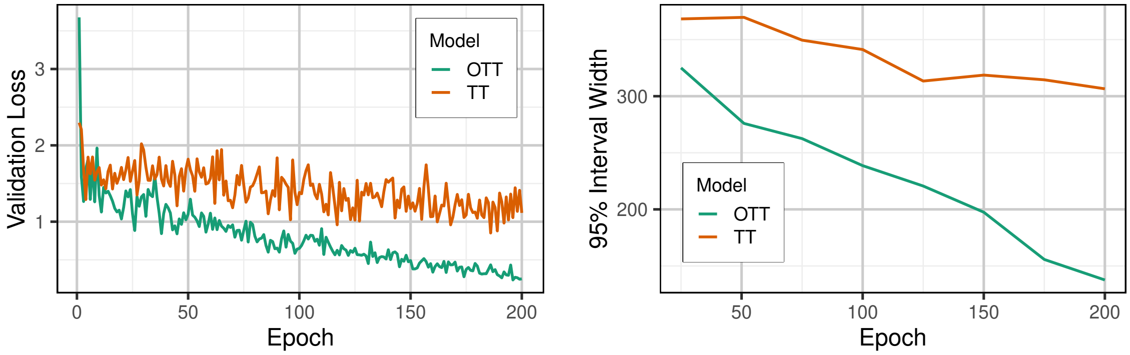


Figure 4.6: Validation losses for reconstruction (left) and confidence interval widths (right) for uncertainty estimation of RAVLT prediction (lower is better).

with different structures can yield direct estimates of uncertainty. Obtaining good measures of this uncertainty requires sampling all parameters a significant number of times: large networks may require many samples before a reasonable uncertainty estimate. Using tensor train constructions allows us to feasibly compute an estimate of uncertainty over all outputs, and with OTT we can further reduce this required sampling rate.

Results. Figure 4.6 (right) shows 95% interval widths computed over 100 MC Dropout instantiations, averaged over individuals in the validation set. The advantage of our compressed orthogonal construction is clear, resulting in tighter confidence intervals compared to the standard TT decomposition.

4.5 Conclusion

Taking advantage of the structure inherent in tensor train decompositions, we propose and analyze the Orthogonal Tensor Train Decomposition, yielding direct benefits in both parameter efficiency and computation time. This is an important step in instantiating recurrent or sequential models for a set of longitudinal 3D brain images, either in the context of generating new images in the sequence or for classification. Using a mapping from Euclidean space, we construct a neural network variable that can efficiently be learned through existing deep learning optimization frameworks. Our results yield promising developments in applying deep learning methods for analyzing sequential 3D medical imaging data, and we show that our method can perform favorably in reconstruction and prediction tasks with such image volumes.

While a focus here was brain imaging, we anticipate numerous applications in other medical imaging settings. Code is available at `https://github.com/ronakrm/OTT`.

Chapter 5

Efficient Learning and Unlearning via Large-Scale Conditional Independence Testing

While selecting a more efficient model *a priori* can be helpful when we can plan ahead of training, it may sometimes be the case that a task may require selection after a model has been trained. In this chapter we will address this post-hoc selection task from a model *parameter* point of view, contrasting our approach from Chapter 4 where we inform a reduced-parameter *model architecture* before training. Here we are motivated by the need for an efficient way of “scrubbing” *existing* models, to address requests for sample deletion. Unlearning in this context has only been analyzed under the assumption that model updates can be made over the *entire* model, but the forms of provable updates have limited practical deployments for feasibility as we will see. Using conditional independence ideas we will identify a **parameter subset** sufficient for efficient unlearning. Work in this chapter was first published in the conference on Computer Vision and Pattern Recognition (Mehta et al., 2022).

5.1 Introduction

As personal data becomes a valuable commodity, legislative efforts have begun to push back on its widespread collection and use, particularly for training ML models. Recently, a focus has been the “right to be forgotten" (RTBF), i.e., the right of an individual’s data to be deleted from a database, and derived products. Despite existing legal frameworks on fair use, industry scraping has led to personal images being used without consent, e.g. Harvey (2021), and even explicit personal information being exposed at inference time in language models (Carlini et al., 2021). While regulation (GDPR, CCPA) has not specified the extent to which data must be forgotten, it poses a clear question: is deletion of the data enough, or does a model trained on that data also needs to be updated?

Work originally presented in Carlini et al. (2019) and Carlini et al. (2020) has identified scenarios where trained models are vulnerable to attacks that can reconstruct input training data. More directly, recent rulings by the Federal Trade Commission FTC (2021); Kaye (2022) have ordered companies to fully delete and destroy not only data, but also any model trained using those data. Public outcry of automatic data collection and use in model training has increased significantly with new results demonstrating the memorization of language models, insofar as providing the full physical address given the proper prompt Carlini et al. (2021). While deletion and (subsequent) full model retraining without the deleted samples is possible, most in-production models require weeks of training and review, with extensive computational/human resource cost. With additional deletions, it is infeasible to retrain each time a new delete request comes in.[1] So, how can we update a model ensuring the data is deleted without retraining?

Task. Given a set of input data $\mathcal{S} : \{z_i\}_{i=1}^n \sim \mathcal{D}$ of size n, training simply identifies a hypothesis $\hat{w} \in \mathcal{W}$ via an iterative scheme $w_{t+1} = w_t - g(\hat{w}, z')$ until convergence, where $g(\cdot, z')$ is a stochastic gradient of a fixed loss function. Once a model at convergence is found, *machine unlearning* aims to identify an update to $\hat{w}$ through an analogous *one-shot unlearning update*:

$$w' = \hat{w} + g_{\hat{w}}\left(z'\right), \tag{5.1}$$

for a *given* sample $z' \in \mathcal{S}$ that is to be **unlearned**.

Contributions. We address several computational issues with existing approximate formulations for unlearning by taking advantage of a new statistical scheme for

[1] Recent large models require upwards of hundreds of thousands to millions of dollars. Preliminary training of Stable Diffusion cost approximately $600,000 (https://twitter.com/emostaque/status/1563870674111832066), and estimates for training GPT-3 range within the millions (https://lambdalabs.com/blog/demystifying-gpt-3).)

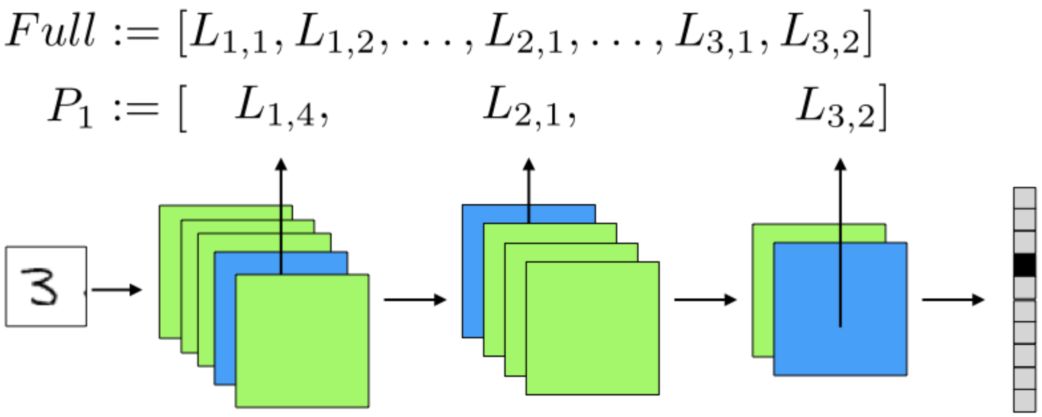

$Full := [L_{1,1}, L_{1,2}, \ldots, L_{2,1}, \ldots, L_{3,1}, L_{3,2}]$
$P_1 := [\ L_{1,4},\ L_{2,1},\ L_{3,2}]$
3

Figure 5.1: Large deep learning networks typically associate specific subsets of network parameters, blocks (blue), to specific samples in the input space. Traditional forward or backward passes may not reveal these blocks: high correlations among features may not distinguish important ones. Input perturbations can be used to identify them in a probabilistic, distribution-free manner.

sufficient parameter selection. First, in order to ensure that a sample's impact on the model predictions is minimized, we propose a measure for computing conditional independence, L-CODEC, which identifies the Markov Blanket of parameters to be updated. Second, we show that the L-CODEC identified Markov Blanket enables unlearning in previously infeasible deep models, scaling to networks with hundreds of millions of parameters. Finally, we demonstrate the ability of L-CODEC to unlearn samples and entire classes on networks, from CNNs and ResNets to transformers, including models for face recognition and person re-identification.

5.2 Problem Setup for Unlearning

Let $\mathcal{A}$ be an algorithm that takes as input a training set $\mathcal{S}$ and outputs a hypothesis $w \in \mathcal{W}$, defined by a set of d parameters Θ. An unlearning scheme $\mathcal{U}$ takes as input a sample $z' \in \mathcal{S}$ used as input to $\mathcal{A}$, and ideally, outputs an **updated** hypothesis $w' \in \mathcal{W}$ where z' has been deleted from the model. An unlearning algorithm should output a hypothesis that is close or equivalent to one that would have been learned had the input to $\mathcal{A}$ been $\mathcal{S} \setminus z'$. A framework for this goal was given by Ginart et al. (2019) as,

Definition 5.1 ($(\epsilon, \delta)-$ forgetting)**.** *For all sets $\mathcal{S}$ of size n, with a "delete request" $z' \in \mathcal{S}$,*

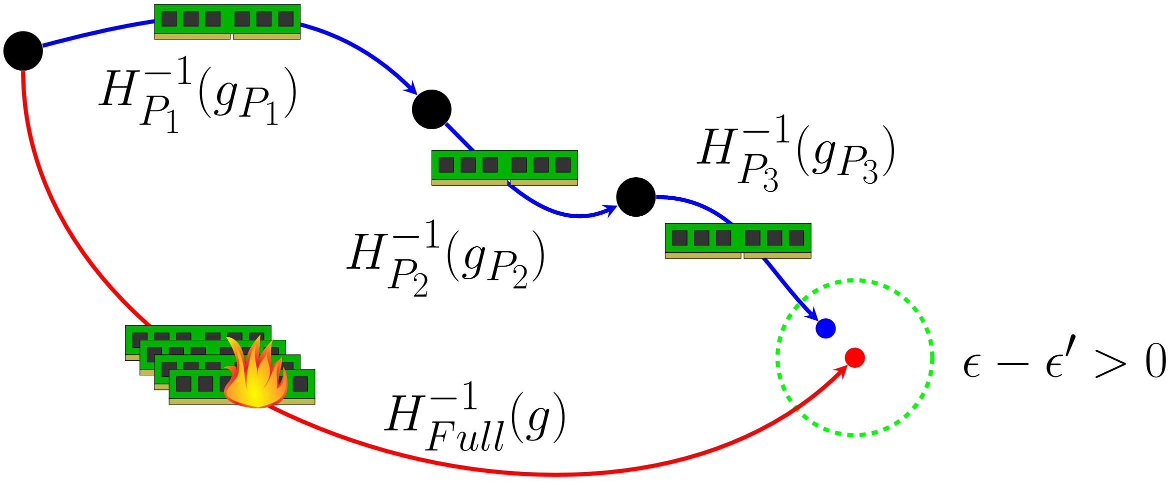


Figure 5.2: Network blocks can be unlearned together in an efficient block-coordinate style update (blue lines), approximating an update to the full network which requires a costly, and sometimes infeasible due to memory constraints, full Hessian inverse (red line).

an unlearning algorithm $\mathcal{U}$ is $(\epsilon, \delta)-$forgetting if

$$\mathbb{P}(\mathcal{U}(\mathcal{A}(S), z') \in \mathcal{W}) \leq e^{\epsilon}\mathbb{P}(\mathcal{A}(\mathcal{S} \setminus z') \in \mathcal{W}) + \delta \tag{5.2}$$

In essence, for an existing model w, a good unlearning algorithm for request $z' \in \mathcal{S}$ will output a model $\hat{w}$ close to the output of $\mathcal{A}(S \setminus z')$ (retraining without that sample) with high probability.

Remark 5.2. *Definition 5.1 is similar to the standard definitions of differential privacy. The connection to unlearning is: if an algorithm is $(\epsilon, \delta)-$forgetting for unlearning, then it is also differentially private.*

If $\mathcal{A}$ is an empirical risk minimizer (ERM) for the loss f, let

$$\mathcal{A} : (\mathcal{S}, f) \rightarrow \hat{w}, \tag{5.3}$$

$\hat{w} = \arg\min F(w)$ and $F(w) = \frac{1}{n}\sum_{i=1}^{n} f(w, z_i)$. Recall $g(z')$ from (5.1): our unlearning task essentially involves identifying the form of $g(z')$ for which the update in (5.1) is (ϵ, δ)-forgetting. If an oracle provides this information, we have accomplished the unlearning task.

The difficulty, as expected, tends to depend on f and $\mathcal{A}$. Recent unlearning results have identified forms of f and $\mathcal{A}$ where such a $g(z')$ exists. The authors in Sekhari et al. (2021) define $g(z') = \frac{1}{n-1} H'^{-1} \nabla f(\hat{w}, z')$, where

$$H' = \frac{1}{n-1} \left(n \nabla^2 F(\hat{w}) - \nabla^2 f(\hat{w}, z') \right), \tag{5.4}$$

with additive Gaussian noise $w' = w' + N(0, \sigma^2)$ scaling as a function of n, ϵ, δ, and the Lipschitz and (strong) convexity parameters of the loss f. We can interpret the update using (5.4) from the optimization perspective as a trajectory "reversal": starting at a random initialization, the first order (stochastic gradient) trajectory of w *with* z' is reversed using *residual* second order curvature (Hessian) information at the optimal $\hat{w}$ in (5.4), achieving unlearning. This is shown to satisfy Definition 5.1, and only incurs an additive error that scales by $O(\sqrt{d}/n^2)$ in the gap between $F(w')$ and the global minimizer $F(w^*)$ over the ERM $F(\hat{w})$.

Rationale for approximate schemes. The aforementioned update requires storing, computing, and inverting the $O(d^2)$ Hessian matrix. For a practitioner interested in unlearning, this can only be directly instantiated if one has extensive computational resources. In settings where it is not directly possible to compute the Hessian inverse necessary for $H'^{-1} \nabla f(\hat{w}, z')$, we must consider alternatives.

A potential idea. Our goal is to identify a form of $g(z')$ that **approximates** the Hessian-scaled gradient $H'^{-1} \nabla f(\hat{w}, z')$. Let us consider the Newton-style update suggested by (5.4) as a smoothing of a traditional first order gradient step. The inverse Hessian is a weighting matrix, appropriately scaling the gradients based on the second order difference between the training set mean point $F(\hat{w})$ and the sample of interest $f(\hat{w}, z')$. This smoothing can also be viewed from an information perspective: the Hessian in this case corresponds to a Fisher-style information matrix, and its inverse as a conditional covariance matrix (Golatkar et al., 2021, 2020b). It could be possible that, if there are a **specific set of parameters** that have *small gradients* at $f(\hat{w}, z')$, or if the information matrix is zero or small, then we need not consider their effect.

Examples of this intuition in vision. Bau et al. (2017a); Fong and Vedaldi (2018); Sun et al. (2019) and others have shown that models trained on complex tasks tend to *delegate* subnetworks to specific regions of the input space. That is, parameters and functions within networks tend to (or can be encouraged to) act in *blocks.* For example, activation maps for different filters in a trained (converged) CNN model show differences for different classes, especially for filters closer to the output layer. We formalize this observation as an assumption for samples in the training set.

Assumption 5.3. *For all subsets of training samples $S \subset \mathcal{S}$, there exists a subset of trained model parameters $P^* \subset \Theta$ such that*

$$f(S) \perp w^*_{\Theta \setminus P} \mid w^*_P \tag{5.5}$$

Due to the computational issues discussed above, if we could make such a simple/principled selection scheme practical, it may offer significant benefits.

To contextualize our contributions, we briefly review existing proposals for machine unlearning.

Naïve, Exact Unlearning. A number of authors have proposed methods for exact unlearning, in the case where $(\epsilon = 0, \delta = 0)$. SVMs by Romero et al. (2007) and Karasuyama and Takeuchi (2009), Naïve Bayes Classifiers by Cao and Yang (2015), and k-means methods by Ginart et al. (2019) have all been studied. But these algorithms do not translate to stochastic models with millions of parameters.

Approximate Unlearning. With links to fields such as robustness and privacy, we see more developments in approximate unlearning under Definition 5.1. The so-called ϵ-certified removal by Guo et al. (2020) puts forth similar procedures when $\delta = 0$, and the model has been trained in a specific manner. Guo et al. (2020); Izzo et al. (2021) provide updates to linear models and the last layers of networks, and Golatkar et al. (2020b) andGolatkar et al. (2020a) provide updates based on linearizations that work over the full network, and follow-up work by Golatkar et al. (2021) presents a scheme to unlearn under an assumption that some samples will not need to be removed.

Other work has taken alternative views of unlearning, which do not require or operate under probabilistic frameworks (Bourtoule et al., 2021; Neel et al., 2021). These schemes present good guarantees in the absolute privacy setting, but they require more changes to pipelines (sharding and/or aggregating weaker models) and scale unsatisfactorily in large deep learning settings.

5.3 Randomized Markovian Block Coordinate Unlearning

If there exist entries of the vector $g(z') = H'^{-1}\nabla f(\hat{w}, z')$ that we can, through *some* procedure, identify as zero, then we can simply avoid computing such zero coordinates. Not only can we zero out those particular entries in the inverse and the gradient, but we can take advantage of the blockwise inverse to *completely remove those parameters from all computations.* If possible, it would immediately change the complexity from $O(d^3)$ to $O(p^3)$, where $p \ll d$ is the size of the subset of parameters that we know are *sufficient* to update.

Let $P \subseteq \Theta := \{1, \ldots, d\}$ be the index set of the parameters that are "sufficient" to update. A direct procedure may be to identify this subset P with

$$P = \underset{P \in \mathcal{P}(\Theta)}{\arg\min} \, ||\tilde{w} - \tilde{w}_P||, \tag{5.6}$$

where $\mathcal{P}(\Theta)$ is the *power set* of the elements in Θ and $\tilde{w}_P$ is the subset of the parameters we are interested in updating. Note that a simple solution to this problem *does* exist: choosing the $p = |P|$ parameters with the largest change will minimize this distance for typical norms. This can be achieved by thresholding the updates $g(z')$ for $\hat{w}$. However, this *requires computing the full update for* $g(z')$. We want a preprocessing procedure that performs the selection *before* computation of $g(z')$ is needed.

A probabilistic angle for selection. We interpret a deep network $\mathcal{W}$ as a functional on the input space $\mathcal{D}$. This perspective is common in statistics for variable selection (e.g., LASSO), albeit used *after* the entire optimization procedure is performed i.e., at the optimal solution. The only difference here is that we use it at approximately optimal solutions as given by ERM minimization. Importantly, this view allows us to identify regions in $\mathcal{W}$ that contain the most information about a query sample z'. We will formalize this intuition using recent results for conditional independence (CI) testing. Finding w_P above should also satisfy

$$z' \perp w_{\Theta \setminus P} \mid w_P \tag{5.7}$$

This CI formulation is well studied within graphical models. Many measures and hypothesis tests have been proposed to evaluate it. The *coefficient of conditional dependence* (CODEC) in Azadkia and Chatterjee (2019), along with their algorithm for "feature ordering", FOCI, at first seems to offer a solution to (5.7), and in fact, can be implemented "as is" for shallow networks.

Using CODEC directly for Deep Unlearning is inefficient. There are two issues: First, when applying CODEC to problems with a very large n with discrete values, the cost of tie-breaking for computing nearest neighbors can become prohibitive (see Appendix B.3 for algorithmic details). Second, z' is not a random variable for which we have a number of instances. We defer discussion of the second issue to Section 5.4, and address the first issue here.

Consider the case where a large number of elements have an equal value. With an efficient implementation using $kd-$trees, identifying the nearest-neighbor as required by CODEC would still require expanding the nodes of all elements with equal value. As an example, if we are looking for the nearest neighbor to a point at the origin and there are a large number of elements on the surface of a sphere centered at the origin, we still require checking all entries and expanding their nodes in the tree, even when we know that they are all equal for this purpose.

Interestingly, this problem has a relatively elegant solution. We introduce a randomized version of CODEC, L-CODEC. For variables A, B, C:

$$T_L := T\left(\tilde{A}, \tilde{B}|\tilde{C}\right), \tag{5.8}$$

where $\tilde{A} = A + N(0, \sigma^2)$, and similarly for $\tilde{B}, \tilde{C}$. This additive noise can simply be scaled to the inverse of the largest distance between any points in the set. By requiring this noise to be smaller than any distance between items in the set, the ranking will remain the same between unique discrete values, and will be perturbed slightly for equal ones. In expectation, this will still lead to the true dependence measure. The noise addition is consistent with the Randomization criterion for conditional independence – for A, B, C in Borel spaces, $A \perp B \mid C$ *iff* $A = h(B, U)$ almost surely, for some measurable function h and uniform random variable $U \sim$ Uniform$(0, 1)$ independent of (B, C) as in Orbanz (2016).

Remark 5.4. *An altered version of this setup also gives us a form of explainability, where we can apply sensitivity analysis to each input feature or pixel and estimate its effect on the output via a similarly randomized version of the Chatterjee rank coefficient $T(A, B)$, proposed by Chatterjee (2020).*

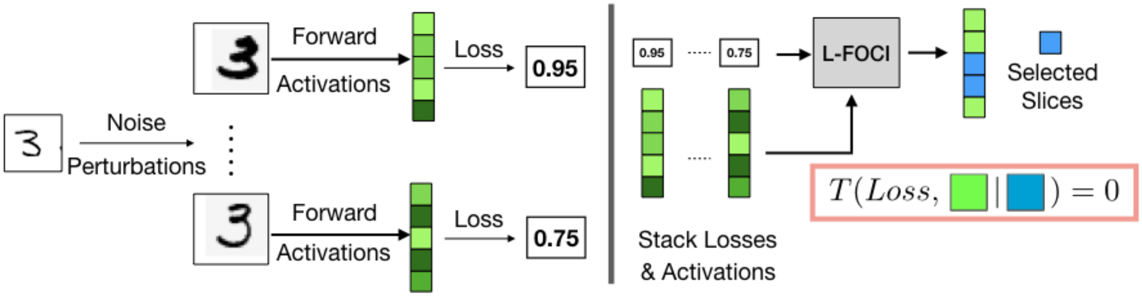


Figure 5.3: A sample is perturbed and passed through the network. Activations are aggregated alongside losses and fed to L-FOCI. Selected rows represent slices of the corresponding layer that are sufficient for unlearning.

Efficient Subset Selection that is also Sufficient for Predictive Purposes

The above test is good for (5.7) if we know *which* subset $P \in \mathcal{P}(\Theta)$ to test. Recent work by Yang et al. (2020) proposes a selection procedure using an iterative scheme to slowly build the sufficient set, adding elements which maximally increase the information explained in the outcome of interest. While it is efficient (polynomial in size), we must know the maximal degree. A priori, we may have no knowledge of what this size is, and for parameter subsets it may be very high.

When using L-CODEC, we can use a more straightforward Markov Blanket identification procedure adapted from Azadkia and Chatterjee (2019). FOCI more directly selects which variables are valuable for explaining z', and in fact, is proven to identify the sufficient set (Markov Blanket) with a reasonable number of samples. Briefly, in our L-FOCI, the sufficient set is built incrementally with successive calls to L-CODEC, moving the most "dependent" feature from the independent set to the sufficient set. See Appendix B.3 for details.

Summary. This procedure alleviates the first issue in terms of sufficient subset or Markov Blanket selection; compared to existing methods using information-theoretic measures that require permutation testing, L-FOCI directly estimates the change in variance when considering a proposal to add to the set. Now, we discuss how this selection can help identify sets of parameters that can be updated.

5.4 Deep Unlearning via L-FOCI Hessians

Our input samples to scrub z' are not random variables for which we have samples or distributional assumptions, nor are our parameters. In this case, a perturbation-based scheme may be useful when attempting to generate samples for unknown distributions.

Considering Assumption 5.3, when only some parameters are useful for the final outcome on an input sample $z' \in S$, the effect of those parameters can be measured through activations due to the forward pass of a model. We estimate the conditional independence test in (5.5) through activations as

$$f(z') \perp a^*_{\Theta \setminus P} \mid a^*_P, \tag{5.9}$$

where a_P for some parameter subset $P \subseteq \Theta$ is defined as the linear activations generated by the forward pass through the model. This formulation relates to a generalized version of the solution in Section 3 of Yang et al. (2020), where conditional mutual information is estimated via feature mappings.

As an example, if a network has linear layers $\mathcal{L}$, a simple linear layer $l \in \mathcal{L}$ with parameters $w_l \in \mathbb{R}^{b_{in} \times b_{out}}$ would have activations $a_l \in \mathbb{R}^{b_{out}}$, with $a_l = w_l a_{l-1}$. For each entry $a_{l,k}$ in the vector a_l, the associated parameters in the layer are $w_l[:, k]$ for $k \in [b]$. Thus, we break up the network into influential *slices*. These slices can be seen as a finer view of the parameter space compared to typical layerwise selection, but coarser than a fully discrete one. Next, $\mathcal{L}$ now refers to the collection of these slices, with a specific slice as l.

The tuple of variables we need samples from is now

$$\{\mathcal{L}(z'), a_1, \dots a_{|\mathcal{L}|}\} \tag{5.10}$$

We can obtain samples from this set by perturbing the input and consecutively collecting activations along all weight slices during the computation of the loss. For a particular perturbation $\xi^j \sim N(0, \sigma^2)$,

$$x_i^j = x_i + \xi^j; \quad l^j, a_L^j = \{l(x_i^j), a_1^j, \dots a_{|\mathcal{L}|}^j\} \tag{5.11}$$

The tuples (l^j, a^j_L) serve as samples for our conditional independence test,

$$(P \subseteq \Theta) = \text{L-FOCI}((l^j, a^j_{\mathcal{L}})^m_{j=1}) \tag{5.12}$$

for m perturbations (see Figure 5.3). This extends naturally to convolution layers, where "slices" correspond to each filter in the layer.

In Algorithm 4, the activations are collected using hooks within the forward pass. First, gradients at the last and penultimate epoch for full training are stored during the original training pass. Given a sample to unlearn, we compute L-FOCI over the perturbed activations and losses generated by the forward pass, and identify which parameter sets will be updated. We compute the approximate Hessian over only these parameters via finite differences for both the full model and for the model only over the sample of interest. Finally, we apply the blockwise Newton update to the subset of parameters as in (5.1) with appropriate noise as in Sekhari et al. (2021).

Algorithm 4: Unlearning via Conditional Dependence Block Selection

```
Data: A trained model $\hat{w}$, gradient vectors $\nabla_1 F(\hat{w}), \nabla_2 F(\hat{w})$, sample $z' \in \mathcal{S}$ to
      unlearn.
Result: model $w'$ with $z'$ removed.
1. for $j \in \{1, \ldots, m\}$ perturbations do
    $\xi^j \sim N(0, \sigma^2)$
    $z'^{,j} = z' + \xi^j$
    $l^j, a^j = f(z'^{,j})$
end
2. Compute $P* = \text{L-FOCI}(l^J, a^J)$.
3. Compute $\nabla^2_P F(\hat{w}, z')$ via finite differences.
4. Update:
    $H'_P = \frac{1}{n-1}\left(n\nabla^2_P F(\hat{w}) - \nabla^2_P f(\hat{w}, z')\right)$   (5.13)
    $w'_P = \hat{w}_P + \frac{1}{n-1} H'^{-1}_P \nabla f(\hat{w}, z')_P$   (5.14)
    $w'_{\Theta \setminus P} = \hat{w}_{\Theta \setminus P}$   (5.15)
```

Computational Gains. A direct observation is that now with this sampling, we add a linear computational load. However, directly updating all parameters requires

$O(d^3)$ computation due to matrix inversion, while this procedure requires $O(md + dm \log m + p^3)$, for the forward passes, FOCI algorithm, and subsequent subsetted matrix inversion. For any reasonable setting, we have $p \ll d$, and so this clearly offers significant practical advantages.

Theoretical Analysis

By definition, any neural network as described above is actually a Markov Chain: we know that the output of a layer is conditionally independent of the penultimate one given the previous one, and clearly a change in one layer will propagate forward through the rest of the network. However, when trained for a task with a large number of samples, the influence or "memory" of the network with respect to a specific sample may not be clear. While the output of the layers may follow a Markov Chain, the parameters in the layers themselves do not, and their influence on a sample through the forward pass may be highly dependent or correlated. Practically, we would hope that unlearning samples at convergence does not cause too much damage to the model's performance on the rest of the input samples. Following traditional unlearning analysis, we can bound the *residual gradient norm* to relieve this tension.

Lemma 5.5. *The gap between the gradient residual norm of the FOCI Unlearning update in Algorithm 4 and a full unlearning update via (5.4),*

$$||\nabla F(w^-_{Foci}, D')||_2 - ||\nabla F(w^-_{Full}, D')||_2 \tag{5.16}$$

shrinks as $O(1/n^2)$, with n the number of training samples.

Proof. The full proof is in Appendix B.1. Main idea: Because we only update a subset of parameters, the gradients for the remainder should not change too much. Any change to a selected layer only propagates to other layers by $1/n$, and a Taylor expansion about the new activation for that layer gives the result. □

How does L-CODEC achieve acceleration for Unlearning? Sampling with weights proportional to the Lipschitz constant of individual filters and layers is an established approach in optimization, see Gorbunov et al. (2020). We argue that L-CODEC computes an approximation to optimal sampling probabilities. Under a mild assumption that the sampling probabilities have full support, it turns out that correctness of our approximate selection procedure can be guaranteed for unlearning purposes by using recently developed optimization tools (Gower et al., 2019). By adapting results from Gorbunov et al. (2020), we can show the following, summarizing the main result of our slice-based unlearning procedure.

Theorem 5.6. *Assume that layer-wise sampling probabilities are nonzero. Given unlearning parameters ϵ, δ, the unlearning procedure in Alg 4 is $(\epsilon', \delta')-$forgetting where $\epsilon' > \epsilon, \delta' > \delta$ represent an arbitrary precision (hyperparameter) required for unlearning. Moreover, iteratively applying our algorithm converges exponentially fast (in expectation) w.r.t. the precision gap, that is, takes (at most) $O(\log \frac{1}{\mathbf{g}_\epsilon} \log \frac{1}{\mathbf{g}_\delta})$ iterations to output such a solution where $\mathbf{g}_\epsilon = \epsilon' - \epsilon > 0$ and $\mathbf{g}_\delta = \delta' - \delta > 0$ are gap parameters.*

Our result differs from Nesterov's acceleration: we do not use previous iterates in a momentum or ODE-like fashion; rather, here we are closer to primal-dual algorithms where knowing nonzero coordinates at the dual optimal solution can be used to accelerate primal convergence, see Diakonikolas and Orecchia (2019). Moreover, since our approach is *randomized*, the dynamics can be better modeled using the SDE framework for unlearning purposes, as in Simsekli et al. (2020). Here, we do not compute anything extra, although it is feasible for future extensions.

Remark 5.7. *Our approach to estimate the Lipschitz constant is different from Fazlyab et al. (2019) where an SDP must be solved – quite infeasible for unlearning applications. Our approach can be interpreted as solving a simplified form of the SDP proposed there, when appropriate regularity conditions on the feasible set of the SDP are satisfied.*

A note on convexity. Existing methods for guaranteeing removal and performance depend on models being convex. Practical deep learning applications however involve highly nonconvex functions. The intuitions of unlearning for convex problems **directly apply to nonconvex unlearning** with one more technical assumption: minimizers of the learning problem satisfy Second Order Sufficiency (SOS) conditions. SOS guarantees that $\nabla^2 \hat{F}(\hat{w})$ and $\hat{H}$ are positive semi-definite, and that the parameter update in (5.4) is an *ascent* direction w.r.t. the loss function on U, making unlearning possible. Guarantees for nonconvex unlearning involve explicitly characterizing a subset of SOS points (so-called "basins of attraction" of population loss), i.e., which points gradient descent can converge to, see Section 1.3 in Traonmilin and Aujol (2020). So, will minimizers from first order methods satisfy SOS conditions? Generally, this is not true, e.g., when the Hessian is indefinite, $\hat{H} \not\succeq 0$, the update itself may not be an ascent direction w.r.t. negative of the loss. Here, standard Hessian modification schemes are applicable Wright et al. (1999), leading to subsequent application of Newton's step in Sekhari et al. (2021) with a diagonally modified Hessian.

We fix weight decay during training, acting as ℓ_2 regularization and giving us an approximate λ-strong convexity. We also take advantage of this property to smooth our Hessian prior to inversion, intuitively extending the natural linearization about a strongly-convex function. Interestingly, this exactly matches a key conclusion from Basu et al. (2021): weight-decay heavily affects the quality of the measured influence, consistent with our nonconvexity discussion.

Implementation Details. As we only need a subset of the Hessian, we compute the finite difference among the parameters within the blocks selected. For large models, even subsets of model parameters may lead to large Hessian computations, so we move parameters as needed to the CPU for parameter updates. Pairwise distance computations for CI testing via nearest neighbor are carried out on the GPU (Zeng et al., 2021). Our code, achieves reasonable run-time for unlearning for deep models, e.g., one unlearning step for a person re-identification task on a ResNet50 model with roughly 24M parameters takes about 3 minutes.

5.5 L-FOCI in Generic ML Settings

We begin with understanding the value of L-CODEC and L-FOCI for Markov Blanket Identification and progress to applications in typical unlearning tasks involving large neural networks previously infeasible with existing scrubbing tools. See Appendix B for additional details.

L-CODEC Evaluation. To assess speedup gained in the discrete setting when running L-CODEC, we construct the Markov Blanket for specific attributes provided as side information with the CelebA dataset. Figure 5.4 shows the wall-clock times for Markov Blanket Selection via FOCI and L-FOCI for each attribute.

Markov Blanket Identification. We replicate the experimental setup in Section 5.3 of Yang et al. (2020), where a high dimensional distribution over a ground truth graph is generated, and feature mappings are used to reduce the dimension and map

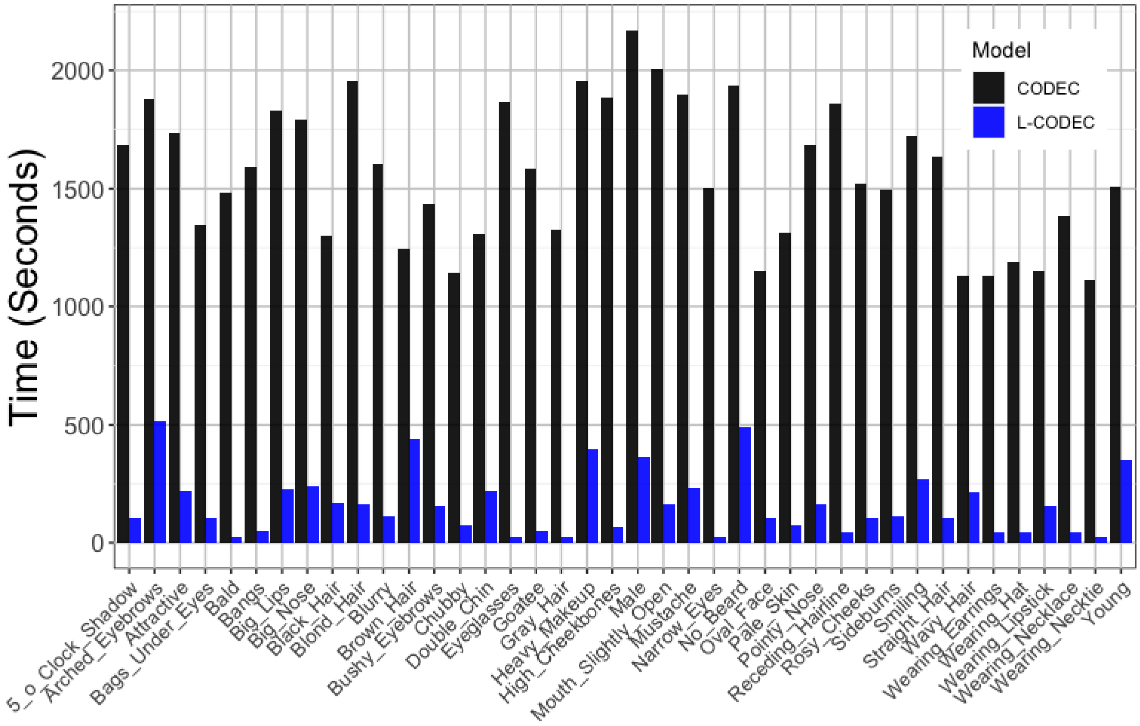


Figure 5.4: L-CODEC vs CODEC run time comparison for identifying sufficient subsets for each CelebA attribute separately (pairs of columns, more details in appendix B.2).

	Raw Data			Feature Maps		
Method	TPR	FPR	Time (s)	TPR	FPR	Time (s)
Yang et al. (2020)	0.75	0.50	5124.22	**0.875**	**0.00**	516.19
L-CODEC + CIT	**1.00**	**0.50**	**402.10**	0.75	**0.00**	117.29
L-CODEC + L-FOCI	N/A			0.833	0.50	**0.464**

Table 5.1: 3D-Bullseye Markov Blanket identification. CIT represents the model in Yang et al. (2020). Both L-CODEC and L-FOCI run much faster than recent Markov Blanket identification schemes. L-FOCI is not applicable to the multi-dimensional raw data setting.

to a latent space. Table 5.1 summarizes subset identification efficacy and runtime. Replacing conditional mutual information (CMI) with L-CODEC, we see a clear improvement in both runtime and Markov Blanket identification over the raw data, and comparable results in the latent feature space. Using L-FOCI directly in the feature space, we identify an additional spurious feature not part of the Markov Blanket, but runtime is significantly faster.

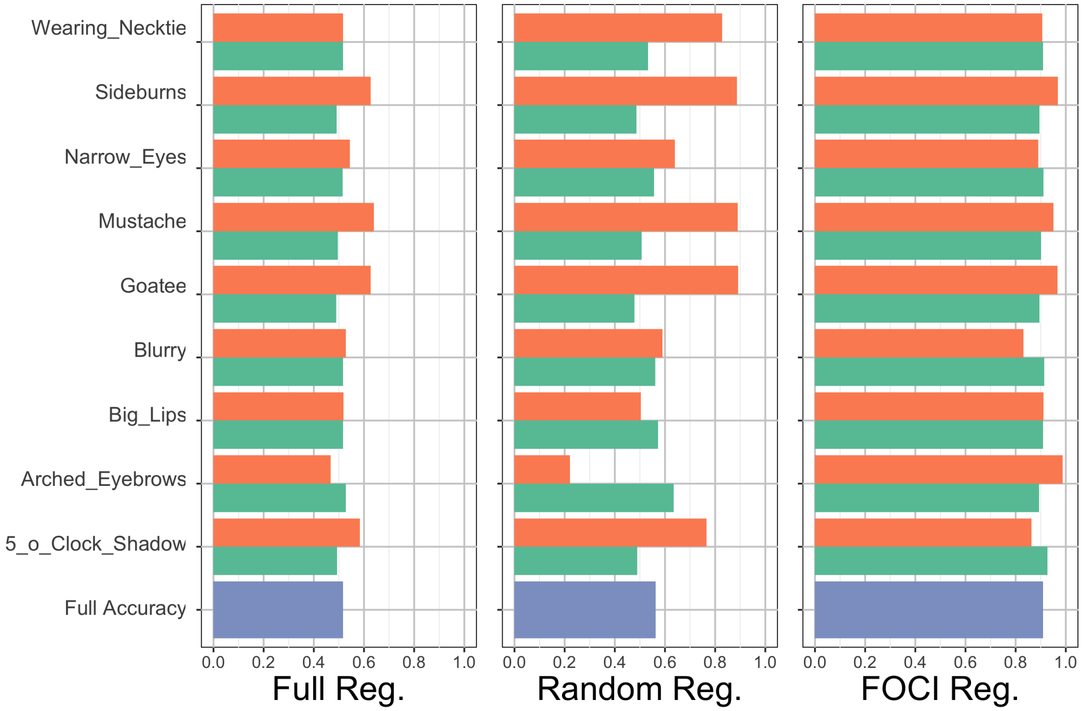


Figure 5.5: Validation accuracies after training to predict "No Beard" in the CelebA dataset. (L to R) regularization for all features, for a random subset, and via FOCI. Green indicates accuracy on the data with that feature, red, without.

Spurious Feature Regularization. This Markov Blanket (MB) identification scheme can be used to address spurious feature effects on traditional NN models. A straightforward approach would be to directly add a loss term for each potentially important feature over which we would like to regularize, $\mathcal{L}(\theta) + \sum_{S \in \mathcal{S}} R_S(\theta)$. However, with a large number of outside factors S, this can adversely effect training. We instead use L-FOCI to identify the set of minimal factors that, when conditioned, make the rest conditionally independent. Then it is only necessary to include regularizers over $S \in MB(Y)$.

We evaluate a simple attribute image classification setting using the CelebA dataset. We run L-FOCI over the attributes as in our L-CODEC evaluation, and regularize using a Gradient Reversal Layer for a simple accuracy term over those attributes. Results in Figure 5.5 clearly show that selection with FOCI provides the best result, maintaining high overall accuracy but also preserving high accuracy on sets of samples with/without correlated attributes.

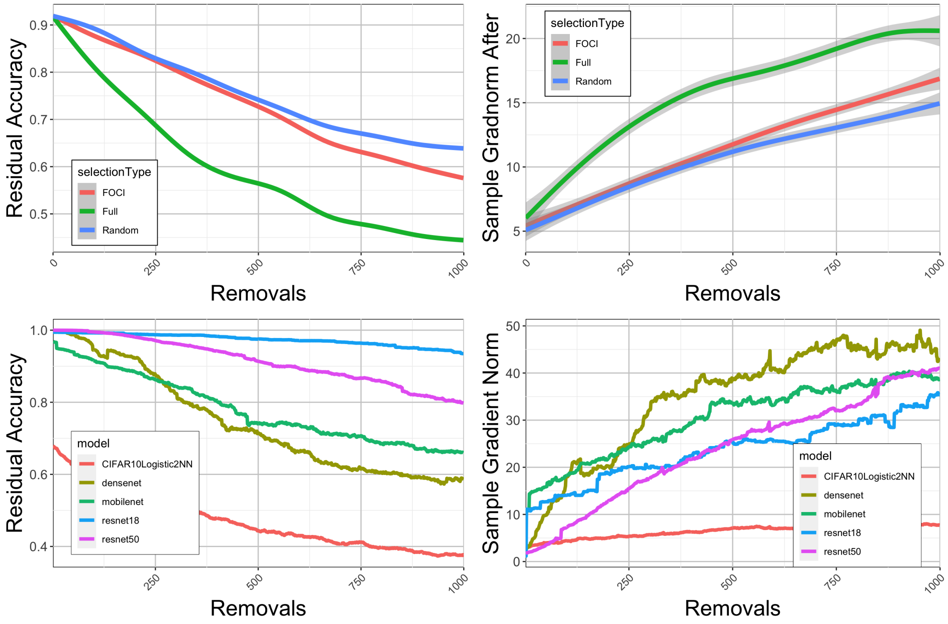


Figure 5.6: (Top) Residual Accuracies & Sample Gradient Norm of removal for an MNIST Logistic Regressor. Averaged over 10 runs. (Bottom) Residual accuracies and sample gradient norms for various CIFAR-10 models.

5.6 L-FOCI for Machine Unlearning

Comparing to Full Hessian Computation

For simple regressors, we can compute the full Hessian and compare results generated by a traditional unlearning update, our L-FOCI update, and a random selection update. To reduce variance and show the best possible random selection, we run our L-FOCI and randomly choose a set of the same size for each random selection. Figure 5.6 (top) shows validation and residual accuracies for 1000 random removals from MNIST (averaged over 10 runs).

Are we selecting reasonable subsets? A natural question is whether the subset selection via L-FOCI is any better than random, given that we are effectively taking a smaller global step. We answer this in the affirmative with a simple comparison with a random selection of size equal to the set selected by L-FOCI. Figure 5.6 (top) shows that the sample gradient norm for selections made by L-FOCI are larger than those of a random selection: the subset of the model scrubbed of this specific sample has a larger impact on its final loss, and thus the gradient norm post-removal is large.

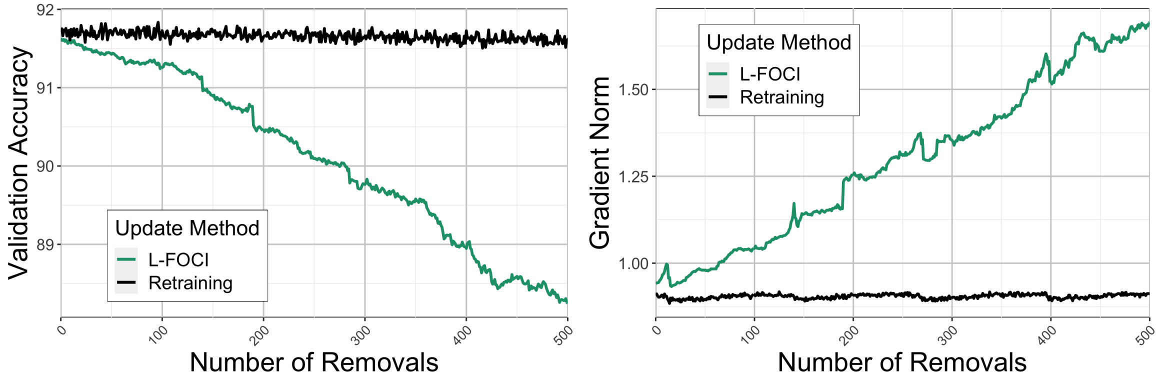


Figure 5.7: MNIST Retraining comparison averaged over 8 runs. Validation accuracies and residual gradient norms.

Does the formulation scale? We scrub random samples from various CIFAR-10 models, and evaluate performance for the same set of hyperparameters. When the models are larger than logistic regression, it is infeasible to estimate the full Hessians, so we *must* use our L-FOCI selection update. Figure 5.6 (bottom) shows removal performance over many typical models with varying sizes. Models that have higher base accuracies tend to support more removals before performance drops. This matches results for differentially private models: models that generalize well may not have overfit and thus may already be private, allowing "fast" forgetting.

Tradeoff vs Retraining. While our focus is the setting in which retraining is not feasible, where we can retrain we compare validation accuracies as a function of number of removals. Using a subset of MNIST, we train to convergence and iteratively remove samples using our construction, retraining fully at each step for comparison. With 1000 training samples from each class and reasonable settings of privacy parameters ($\epsilon = 0.1, \delta = 0.01$), we support a large percentage of removals until validation accuracy drops more than a few percent, see Figure 5.7.

Removal in NLP models

We now scrub samples from transformer based models using LEDGAR (Tuggener et al., 2020), a multilabel corpus of legal provisions in contracts. We use the prototypical subset which contains 110156 provisions pertaining to 13 most commonly used labels based on frequency. Our model is a fine-tuned DistilBERT (Sanh et al., 2019) and uses the $[CLS]$ token as an input to the classification head. Table 5.2 shows

	# Supported Removals	
ϵ	Governing Laws	Terminations
0.1	> 100	> 100
0.01	> 100	> 100
0.001	18	21
0.0005	6	7

Table 5.2: Scrubbing transformer model for provision classification.

results of scrubbing the provisions from two different classes; *Governing Laws* and *Terminations* which have the highest/lowest support in the test set. As expected with increasing ϵ, i.e., lower privacy guarantees, we can support more number of removals based on the Micro F1 score of the overall model. The Micro F1 scores, for the removed class fall off rapidly, while the change in overall scores is more gradual.

Removal from Pretrained Models

The above settings show settings where a sample from one specific source may be removed. A more direct application of unlearning is completely removing samples from a specific class; a compelling use case is face recognition.

We utilize the VGGFace dataset and model, pretrained from the original work in Huang et al. (2008); Parkhi et al. (2015). The model uses a total of approximately 1 million images to predict the identity of 2622 celebrities in the dataset. Using a reconstructed subset of 100 images from each person, we first fine-tune the model on this subset for 5 epochs, and use the resultant models as estimates of the Hessian. In this setting, the VGGFace model is very large, including a linear layer of size 25088×4096. Selecting even a few slices from this layer results in a Hessian matrix unable to fit in typical memory. For this reason, we run a "cheap" version of L-FOCI: we select only one slice that results in the largest conditional dependence on the output loss.

Figure 5.8 show results for scrubbing consecutive images from one individual in the dataset for a strong privacy guarantee of $\epsilon = 10^{-5}$. As the number of samples scrubbed increases, the performance on that class drops faster than on the residual set, exactly as desired.

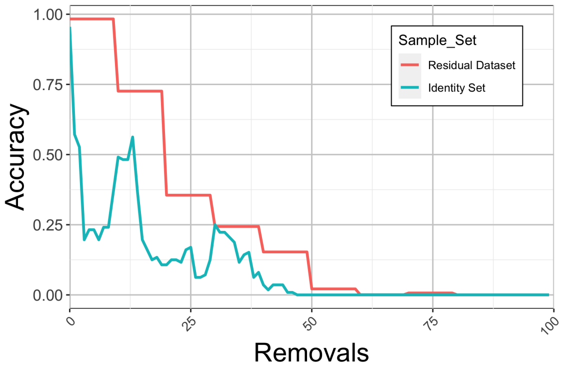


Figure 5.8: Scrubbed and Residual Accuracies (every 10 removals) for $\epsilon = 1e^{-5}$. The accuracy drop for the residual set is gradual up to a certain number of removals.

Removal from a Person re-identification model

As a natural extension to our experiments on face recognition, we evaluate unlearning of deep neural networks trained for person re-identification. Here, the task is to associate the images pertaining to a particular individual but collected in diverse camera settings, both belonging to the same camera or from multiple cameras. In our experiments, we use the Market-1501 dataset (Zheng et al., 2015) and a Resnet50 architecture which was trained for the task. We unlearn samples belonging to a particular person, one at a time, and check the performance of the model. Experimental results are in agreement with results reported for the transformer model as well as the VGGFace model. With very small values of ϵ i.e. 0.0005 the number of supported removals is limited to less than 10 depending on the person id being removed. However, with a larger value of ϵ, e.g., 0.1, all potential samples can be removed without a noticeable degradation in model performance in terms of mAP scores. In Figure 5.9, we clearly see that after scrubbing a model for a particular person, its predictions for that particular individual become meaningless whereas the predictions on other classes are still possible with confidence, as desired. Additional experiments with different datasets, model architectures and other ablations for deep unlearning for person re-identification models are presented in Appendix B.2.

5.7 Conclusion

Our selection scheme identifies a subset of parameters to update and significantly reduces compute requirements for standard Hessian unlearning. For smaller networks

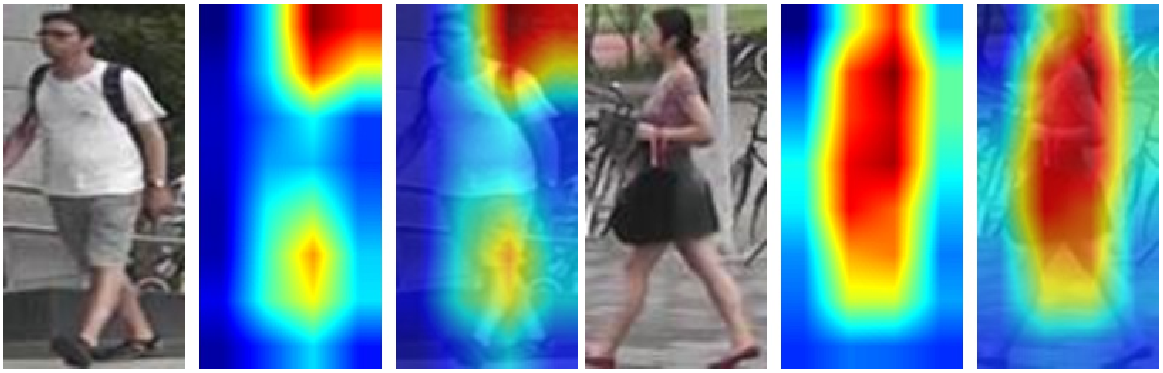


Figure 5.9: Activation maps from a model scrubbed for the person on the left (right set is not scrubbed). For each triplet, from (L to R) are the original image, the activation map and its image overlay. Note the effect of scrubbing: activations change significantly for the scrubbed sample (compare column 2 to 3) whereas remain stable for the non-scrubbed sample (compare column 5 to 6).

with a large number of removals, retraining may be effective, but when full training sets are not available or retraining is costly, unlearning in some form is needed. We show the ability to approximately unlearn for large models prevalent in vision, a capability that has not so far been demonstrated.

Chapter 6

Generalizing the Earth Mover's Distance for Efficient Neural Network Regularization

The unlearning procedures above work well when samples to be removed are known. In cases of data privacy these samples are clear, defined by guidelines or direct user requests. However in the case where individual samples are not explicitly identified, we may still want a model to behave as if it is agnostic to individual samples, or perform equally across different individuals or groups (subsets) of people. Building a model beforehand that has been trained in a manner that automatically provides a similar guarantee can alleviate some of the post-hoc computational stressors described in the last chapter. Here, we will describe a fast method for both **identifying samples during training** that are outliers in a particular sense, and directly selecting and pushing model parameters towards reducing disparate model performance on those samples. Work in this chapter appeared at the International Conference on Learning Representations (Mehta et al., 2023).

6.1 Introduction

The use of Optimal transport (OT) is now prevalent in many problem settings including information retrieval (Balikas et al., 2018; Yurochkin et al., 2019), image processing (Bonneel et al., 2014), statistical machine learning, and more recently, for ethics and fairness research (Kwegyir-Aggrey et al., 2021). OT is well-suited for tasks where dissimilarity between two or more probability distributions must be quantified; its success was made possible through dramatic improvements in algorithms (Cuturi, 2013; Solomon et al., 2015) that allow one to efficiently optimize commonly used functionals. In practice, OT is often used to estimate and minimize the distance between certain (data-derived) distributions, using an appropriately defined loss functional. When one seeks to operate on more than two distributions, however, newer constructions are necessary to effectively estimate distances and transports. To this end, a well studied idea in the literature is the "barycenter," identified by minimizing the pairwise distance between itself and all other distributions given (analagous to the Karcher mean on Riemannian manifolds). The d-dimensional proxy distance is then defined as the sum of the distances to the barycenter.

Computing barycenters. Assuming that a suitably regularized form of the optimal transport loss is utilized, the pairwise distance calculation itself can be efficient – in fact, in some cases, Sinkhorn iterations can be used (Cuturi, 2013). On the other hand, to minimize distances to the mean, most algorithms typically operate by repeatedly estimating the barycenter and those pairwise distances, and using a "coupling" strategy to push points toward the barycenter, or in other cases, summing over all pairwise distances. As the number of distributions grows, robustness issues can exacerbate (Alvarez-Esteban et al., 2008; Le et al., 2021) and the procedure becomes expensive (e.g., for 50 distributions, with 50 "bins").

A potential alternative. Multi-marginal optimal transport (MMOT) is a related problem to the aforementioned task but to some extent, the literature has developed in parallel. In particular, MMOT focuses on identifying a joint distribution such that the marginals are defined by the input distributions over which we wish to measure the dissimilarity. The definition naturally extends the two-dimensional formulation, and recent work has explored a number of applications (Pass, 2015). But the MMOT computation can be quite difficult, and only very recently have practical algorithms been identified (Lin et al., 2022). Additionally, even if a suitable method for computing an analogous measure of distance were available, *minimizing* this distance to reduce dissimilarity (push distributions closer to each other) is practically hard if standard interior point solvers are needed just to compute the distance itself.

Why and where is dissimilarity important? Enforcing distributions to be similar is a generic goal whenever one wishes some outcome of interest to be agnostic about particular groups within the input data. In applications where training deep neural network models is needed, it is often a goal to enforce distribution similarity on model outputs. For example, in Jiang et al. (2020), the authors define fairness measures over the probability of the prediction, given ground truth labels. However, these methods are rarely extended to continuous measures among internal neural network activations, mainly due to the strong distributional assumptions needed (product of Gaussians) and the added algorithmic complexity of estimating the barycenter. These issues limit application of these ideas to only the final outputs of neural network models, where the distribution is typically binomial or multinomial. MMOT solutions might be employed here, but suffer similar computational limitations.

Contributions. **(1)** We identify a particular form of the discrete multi-marginal optimal transport problem which admits an extremely fast and numerically robust solution. Exploiting a recent extension of the classical Earth Movers Distance (EMD) to a higher-dimensional Earth Mover's objective, we show that such a construction is equivalent to the discrete MMOT problem with Monge costs. **(2)** We show that minimization of this *global* distributional measure leads to the harmonization of input distributions very similar in spirit to the minimization of distributions to barycenters (see Figure 6.1). **(3)** We prove theoretical properties of our scheme, and show that the gradient can be read directly off from a primal/dual algorithm, alleviating the need for computationally intense pairwise couplings needed for barycenter approaches. **(4)** The direct availability of the gradient enables a specific neural network instantiation, and with a particular scaffolding provided by differentiable histograms, we can operate directly on network activations (anywhere in the network) to compute/minimize the d-MMOT. We establish via experiments that computing gradients used in backpropagation is fast, due to rapid access to solutions of the dual linear program. We compare with barycenter-like approaches in several settings, including common fairness applications.

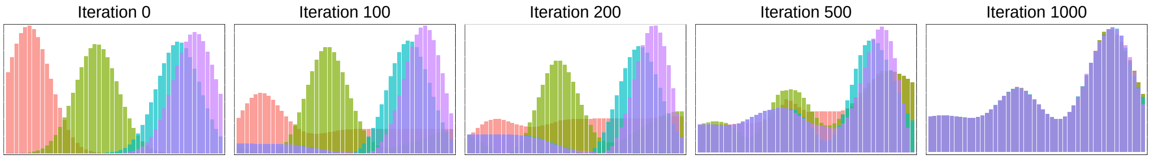


Figure 6.1: Starting and ending state of minimizing a multi-marginal OT distance. Each iteration minimizes the generalized Earth Mover's objective, and then updates each histogram in the direction provided by the gradient.

6.2 Existing Work on Optimal Transport and Related Work

Despite originating with (Monge, 1781), optimal transport continues to be an active area of research (Villani, 2009). The literature is vast, but we list a few key developments.

Early applications. Starting in (Peleg et al., 1989), the idea of shifting "mass" around within an image was used for comparing images to each other and applied to image retrieval (Rubner et al., 2000), where the term "Earth Mover's Distance" (EMD) was introduced. EMD has since been widely used in computer vision: e.g., for image warping (Zhang et al., 2011), in supervised settings (Wang and Guibas, 2012), matching point sets (Cabello et al., 2008) and in scenarios involving histogram comparisons (Ling and Okada, 2007; Wang and Guibas, 2012; Haker et al., 2004).

Modern machine learning. The continuous optimal transport problem (Monge-Kantorovich problem), was originally presented in (Kantorovich, 1942; Kantorovitch, 1958). While the continuous problem has been studied intensively (Villani, 2021), uses of optimal transport within machine learning were possible due to (Cuturi, 2013), which showed that entropic regularization enables fast algorithms for EMD (two distributions with discrete support), and contributed to the success of Wasserstein distances in applications. Consequently, problems including autoencoders (Tolstikhin et al., 2018), GANs (Arjovsky et al., 2017), domain adaptation (Courty et al., 2016), word embeddings (Huang et al., 2016) and classification tasks (Frogner et al., 2015) have benefited via the use of optimal transport.

Multi-marginal optimal transport. Extending optimal transport theory to an *arbitrary number* of distributions has been studied on the theoretical side (Pass, 2015), and practical extensions have been proposed (Lin et al., 2022). But implementations that integrate directly into machine learning pipelines rely on either heuristics or modifications that result in an approximation of the original MMOT problem (Cao et al., 2019).

Wasserstein barycenters. One use case of our algorithms will be in formulations that involve Wasserstein barycenters. Given a set of probability distributions, the Wasserstein barycenter minimizes the *mean* of Wasserstein distances to *each* probability distribution in the set: a practical definition of the mean under the transportation distance (Luise et al., 2019; Cuturi and Doucet, 2014; Agueh and Carlier, 2011; Janati et al., 2020). Applications of Wasserstein barycenter include texture analysis (Rabin et al., 2011), sensor data fusion (Elvander et al., 2020), shape interpolation (Solomon et al., 2015), coupling problems (Rüschendorf and Uckelmann, 2002) and others (Ho et al., 2017). Very recently, polynomial time algorithms have been derived (Altschuler and Boix-Adsera, 2021).

Fairness. Proposals such as (Jiang et al., 2020; Chzhen et al., 2020; Gordaliza et al., 2019) have all regularized models towards outcomes which have equal predictive power over subgroups within a population, measured using optimal transport distance. Informally, the idea is to operate on distributions that are supported on disparate groups, and ask that the distributions get “pushed" towards a common central distribution. This requires solving an optimal transport problem.

Greedy algorithms and extensions. Hoffman et al. (1963) observed that there exists a family of linear-time greedy algorithms that solve the classical two-dimensional transportation problem. Later, Bein et al. (1995) extended the relevant definitions and the greedy algorithm to d-dimensional transportation problems. More recently, the results in Kline (2019), which we will use here, further extended this result to the dual program, and several theoretical properties of the generalized Earth Mover's problem were shown. A slightly different generalized d-dimensional Earth Mover's problem is explored in Erickson (2020), with a focus on statistical generalization properties.

6.3 Earth Mover's Distance and Discrete Multimarginal Optimal Transport

Given a pair of discrete probability distributions $p_1, p_2 \in \mathbb{R}^n_+$, we may want to quantify similarity or dissimilarity. Often we do this by selecting from many measures, including the q-norm, KL-divergence or the Earth Mover's Distance (EMD). The EMD for a pair of distributions has several equivalent interpretations. First, let p_1 be a source of mass, and p_2 be a sink for mass, and $x(i,j)$, where $x \in \mathbb{R}^{n\times n}$, represent the flow of mass from $p_1(i)$ to $p_2(j)$. Denote by $c(i,j)$ the cost of moving one unit of mass from $p_1(i)$ to $p_2(j)$. The EMD between p_1 and p_2 is the minimal cost to transform p_1 into p_2, written as a linear program (LP):

$$\min_{x\in\mathbb{R}^{n\times n}_+} \sum_{i,j} c(i,j)x(i,j) \quad \text{s.t.} \quad \sum_j x(i,j) = p_1(i); \; \sum_i x(i,j) = p_2(j), \; (\forall i,j \in [n]). \tag{6.1}$$

The source-sink interpretation is asymmetric in p_1 and p_2, but the LP is symmetric in p_1 and p_2. It can be shown that the objective value of this LP defines a *metric* (Kantorovich, 1960), and the optimal value of the objective function can be interpreted as a distance between p_1 and p_2, and useful to quantify dissimilarity between pairs of distributions. In particular, $p_1 = p_2$ if and only if the optimal objective value of the Earth Mover's problem vanishes. The LP in (6.1) has an equivalent dual LP,

$$\max_{z_1,z_2\in\mathbb{R}^n} z_1^\top p_1 + z_2^\top p_2 \quad \text{s.t.} \quad z_1(i) + z_2(j) \le c(i,j), \; (\forall i,j \in [n]). \tag{6.2}$$

By strong duality, the optimal value of the primal program (6.1) equals the optimal value of the dual program (6.2). Many practical relaxations have been proposed for (6.1), including entropic regularization (Cuturi, 2013). Computation of the EMD is readily available as in the Python Optimal Transport (POT) library (Flamary et al., 2021).

Discrete Multi-Marginal Optimal Transport

The foregoing approach applies only to $d = 2$ distributions, namely p_1 and p_2. We briefly review the extension to $d > 2$ distributions; the literature calls this *multi-marginal optimal transport* (*MMOT*).

Definition 6.1 (Discrete Multi-Marginal Optimal Transport (d-MMOT)). *Let $p_1, \ldots, p_d \in \mathbb{R}^n_+$ be discrete probability distributions. Let $C_d : \mathbb{R}^{n^d} \to \mathbb{R}_+$. The discrete multi-marginal optimal transport problem (d-MMOT) can be written as*

$$\mathop{\mathit{min}}_{X \in \mathbb{R}^{n \times \cdots \times n}} \quad C_d(X) \quad \textit{s.t.} \quad X_i = p_i, \ (\forall i \in [d]),$$

where $X_i \in \mathbb{R}^n$ is the i-th marginal of $X \in \mathbb{R}^{n \times \cdots \times n} = \mathbb{R}^{n^d}$.

Following the original formulation (Kantorovich, 1942), we will restrict the cost function $C_d(\cdot)$ to the linear map, $C_d(X) := \langle c, X \rangle_\otimes$, where $c \in \mathbb{R}^{n \times \cdots \times n}_+$ is nonnegative. Here, the d-MMOT is the LP,

$$\begin{aligned} \min_{x \in \mathbb{R}^{n^d}_+} \sum_{i_1,\ldots,i_d} c(i_1,\ldots,i_d)\, x(i_1,\ldots,i_d) \quad \text{s.t.} \sum_{i_2,\ldots,i_d} x(i_1,\ldots,i_d) &= p_1(i_i), (\forall i_1 \in [n]) \\ &\vdots \\ \sum_{i_1,\ldots,i_{d-1}} x(i_1,\ldots,i_d) &= p_d(i_d), (\forall i_d \in [n]). \end{aligned} \tag{6.3}$$

This linear program is central to the regularization schemes that are discussed below.

6.4 Efficient d-MMOT Computation

The linear d-MMOT problem (6.3) suffers from the curse of dimensionality: the LP has n^d variables, and even modest choices of n and d can result in a LP with billions of variables, making standard LP solvers inapplicable. Alternatively, specific algorithms have been proposed (Benamou et al., 2015), and relaxations via entropic regularization have become more widespread, with very recent extensions to the d-MMOT setting (Tupitsa et al., 2020; Lin et al., 2022).

In practice, the cost c in (6.3) typically takes one of two forms. In the case where the distributions $p_1, \ldots, p_d$ are over categorical variables, the cost is typically defined as $c(i_1, \ldots, i_d) = 0$ when $i_1 = \cdots = i_d$ and 1 otherwise. However, and importantly, if the distributions p_i are histograms over some ordinal or discretized space, the cost typically has a structure closer to that of a "tensorized" distance, characterized by the *Monge* property.

Definition 6.2 (Monge Property). *A tensor c is Monge if for all valid $i_1, \ldots i_d$ and*

$j_1, \ldots, j_d$,

$$c(s_1, \ldots, s_d) + c(t_1, \ldots t_d) \leq c(i_1, \ldots i_d) + c(j_1, \ldots, j_d) \tag{6.4}$$

where $s_k = \min(i_k, j_k)$ *and* $t_k = \max(i_k, j_k)$.

Our focus is on a specific cost, which is known to be Monge: $c(i_1, i_2, \ldots, i_d) \coloneqq \max\{i_k : k \in [d]\} - \min\{i_k : k \in [d]\}$. When $d = 2$, this cost reduces to $c(i_1, i_2) = |i_1 - i_2|$, which agrees with the classical EMD cost. This choice of c is called the *generalized EMD cost*.

Remark 6.3. *Limiting our attention to this cost is not as restrictive as it may appear. Indeed, Bein et al. (1995) shows that the optimal solution to the LP (6.3) is independent of the cost, as long as it is Monge. Additionally, when c is the generalized EMD cost, Kline (2019) describes a greedy algorithm that solves both (6.3) and its dual (6.5) in linear time. It is also shown that the optimal objective value is a continuous nonnegative function of each probability distribution p_j (i.e., small changes in one distribution cause small changes in the objective value). Continuity is critical for numerical stability. Next, when c is the generalized Earth Mover's array, the optimal objective value vanishes if and only if $p_i = p_j$ for all $i, j \in [n]$.*

This *separability* property is useful in applications where we wish to iteratively "step towards" the barycenter of a set of distributions, see Figure 6.2. In order to step towards red arrows) a barycenter, we require a descent direction. We observe that the following result gives us this functionality.

Theorem 6.4. *The dual linear program of the d-MMOT problem (6.3) is*

$$\underset{z_j \in \mathbb{R}^n, j \in [d]}{\textit{maximize}} \quad \sum_j p_j' z_j \quad \textit{subject to} \quad z_1(i_1) + \cdots + z_d(i_d) \leq c(i_1, \ldots, i_d), \tag{6.5}$$

where the indices in the constraints include all $i_j \in [n]$, $j \in [d]$. Denote by $\phi(p_1, \ldots, p_d)$, the optimal objective value of the LP in (6.3). Let z^* be an optimal solution to the dual program (6.5). Then,

$$\nabla \phi(p_1, \ldots, p_d) = z^*, \quad \textit{and for any } t \in \mathbb{R}, \;\; \phi(p_1, p_2, \ldots, p_d) = \sum_j p_j'(z_j^* + t\,\eta),$$

where $\eta \coloneqq (z_1^*(n)\, e, z_1^*(n)\, e, \cdots, z_d^*(n)\, e)$.

Proof. The main observation invokes perturbation analysis (Mangasarian and Meyer, 1979; Ferris and Mangasarian, 1991) of LPs to assert that, under mild uniqueness conditions, small changes to a LP's input data does not change its optimal solution. The full proof is in Appendix C.1. □

Remark 6.5. *The first part of this result provides what we require: a direction of descent. Thus, if we can solve the d-MMOT problem and also find the optimal solution to its dual, then we can step (or move) our distributions in the opposite direction of the dual variables to push them together, see Figure 6.2. The second claim is somewhat technical, and reconciles particular affine shifts that result in equivalent objective values.*

Optimization of d-MMOT

Setup. We can now instantiate d-MMOT as an add-on term in a standard machine learning formulation. Concretely, it can be positioned alongside typical learning losses $L(f(x), y; \theta)$ to encourage minimizing distances among d different distributions $g_i \in G, i \in [d]$, d-MMOT$(f(x), g; \theta)$, i.e.,

$$\min_{\theta} L(f(x), y; \theta) + \text{d-MMOT}(f(x), g; \theta). \tag{6.6}$$

Within a deep neural network (DNN) architecture, as in some of our experiments, several properties of the d-MMOT module are useful: our results above naturally provide clean operations for computing both the required objective in the "forward" pass and gradients in the "backward" pass.

Using the primal and dual variables. If the optimal objective value of (6.3) serves in regularizing a deep neural network, then we can train the network as follows. During the forward pass, i.e., computing the d-MMOT objective, we can employ a version of the aforementioned primal/dual algorithm that solely computes the function ϕ and stores the dual variables z. As backpropagation proceeds, when the EMD module encounters an incoming gradient, it is simply multiplied by the stored dual variables (see Theorem 6.4 and Algorithm 5). We call our procedure the d-dimensional Earth Mover's Distance, or in short, *the DEMD algorithm*.

Complexity. Computing the DEMD distance in the forward pass is exactly $O(nd)$: linear in the number of distributions and number of bins. This property follows

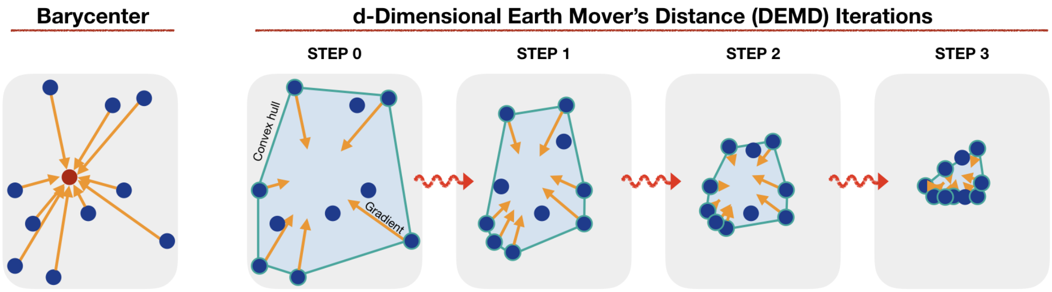


Figure 6.2: (*Left*) Barycenter methods identify a center (red circle) and transport *all* distributions (blue circles) toward that center along the coupling path (yellow). (*Right*) Our DEMD approach identifies "support" distributions that lie on the convex hull (outlined circles), and only those distributions are moved in a direction that decreases the Generalized EMD objective.

Algorithm 5: d-Dimensional Earth Mover's Distance (DEMD)

```
Function Forward($p_1, \ldots, p_d$):
    Compute $\sum s_k t_k$ and $(z_1, \ldots, z_d)$ via Kline (2019).
    Save $(z_1, \ldots, z_d)$.
    return $\sum_k s_k t_k$
Function Backward($gradOutput$):
    Load $(z_1, \ldots, z_d)$
    return $(z_1, \ldots, z_d) \cdot gradOutput$
```

directly from the algorithm, needing only a single pass through all of the data. Bein et al. (1995) provides this result for greedy algorithms that solve OT programs as in Definition 6.1. In contrast to methods that derive gradients via entropic regularization schemes, i.e., relaxations of the optimal transport problem (Luise et al., 2018; Xie et al., 2020; Cuturi et al., 2020), this approach solves the distance computation exactly in linear time. This analysis is not only provided by the theory in prior work, but is also explicit in the number of iterations defining our algorithm (see Appendix C.4 for more details). For minimizing the DEMD (computing the updates, i.e., red arrows in Figure 6.2), a convergence analysis would follow from the properties of the optimization scheme chosen. Our tool can be dropped in exactly as any other module in modern learning applications (using the observation that gradients are easily computed, i.e., $O(1)$ time to read stored dual variables).

A few practical adjustments. It often happens during training that the optimal solutions may, through updates by stepping in the direction of the gradient, acquire entries that are negative. This violates an assumption that entries must be nonnegative. However, Theorem 2.2 in Kline (2019) shows that the optimal objective value, ϕ, possesses a type of translation invariance. We leverage this result, alongside the second part of our Theorem 6.4 to ensure that, in the event that a point escapes the nonnegative orthant of $\mathbb{R}^n$, an appropriately constructed constant vector may be added to the current iterate so that it again lies in the nonnegative orthant, *without changing the objective value*. Further, with a relatively small learning rate, by the continuity of ϕ, normalization is a small perturbation of the original point and can be applied as necessary to enforce that updates result in valid distributions.

Remark 6.6. *Another useful property is that if the convex hull of $(p_1, \ldots, p_d)$ is contained in the convex hull of $(\hat{p}_1, \ldots, \hat{p}_{\hat{d}})$, then $\phi(p_1, \ldots, p_d) \leq \phi(\hat{p}_1, \ldots, \hat{p}_{\hat{d}})$, with strict inequality if containment is strict. A direct consequence of this property is that points within the interior of the convex hull of the data fed to the optimization model have vanishing gradients. Practically, this leads to **sparse** gradients w.r.t. the distributions, and nonzero gradients correspond to points on the hull, i.e., distributions which are maximally different from the rest. Minimization proceeds by iteratively moving mass such that these maximally different distributions are pushed toward each other. Figure 6.2 shows the convex hull during minimization of the generalized EMD objective via our DEMD algorithm.*

Need for Histogramming. Note that outputs $f(x)$ and intermediate activations are *continuous* values (layer shape by batch size). So, we must transform activations into normalized histograms (i.e., discrete distributions). While libraries such as PyTorch and Tensorflow typically provide histogram functions, they are not differentiable. The discontinuous operation of bin assignment does not allow for an end-to-end pipeline where we can push upstream parameters in the direction of minimizing the EMD objective over histograms. To address this, we use a simple relaxed/differentiable histogramming operation. First, all outputs are mapped to $[0, 1]$ using a Sigmoid activation. Then, for each bin location, the “count” in that bin is determined by a ReLU function applied to the difference between the gap between the activation and the distance to the bin boundary. In other words, if an activation falls in a bin, the count of that bin increases based on the distance to the bin boundary, otherwise the count remains the same. The full procedure is detailed in Appendix C.2. The ReLU activations defining bin boundaries allow for gradients to move samples towards neighboring bins as needed.

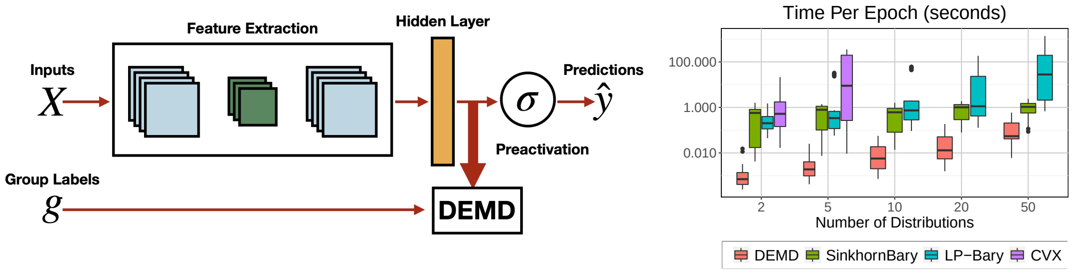


Figure 6.3: (Left) DEMD is positioned after the final layer, prior to the activation. Activations are sorted into distributions utilizing group labels provided alongside the input data. Computed distributions are then brought together using our algorithm. (Right) Computation times for direct distance evaluation of EMD-like distances. Existing methods take much more time (check y-axis) as the number of distributions grow.

6.5 Numerical Evaluations and Fairness Experments

We evaluate our construction in a number of settings. First, we demonstrate the computational speedup associated with evaluating d-MMOT using our algorithm, along with speedups associated with directly computing the gradient. Next, we compute the construction in a series of neural network tasks associated with ensuring distributional similarity: fairness, invariant representations, and multi-domain matching. We provide a complete PyTorch Network Module that packages the above differentiable DEMD objective and histogram functions, and serves as a plug-and-play regularization module.

Performance Benchmarks

Figure 6.4 presents NumPy and Torch instantiations of both forward and backward passes using automatic differentiation and the gradients computed using the dual as in Section 6.4. As expected, the forward (distance computation) times are comparable, but the backwards computation scales poorly with the number of distributions to be updated using automatic differentiation. Our dual setting allows the gradients to simply be read off (based on the forward pass), leading to **no** additional computation overhead during backpropagation.

We compare our DEMD computation to what one may use given existing Optimal Transport methods in Figure 6.3 (right). Using standard off-the-shelf methods as a baseline, when the cost is Monge our algorithm provides *significant* speedups in computation time, on the scale of orders of magnitude! Further, if the number

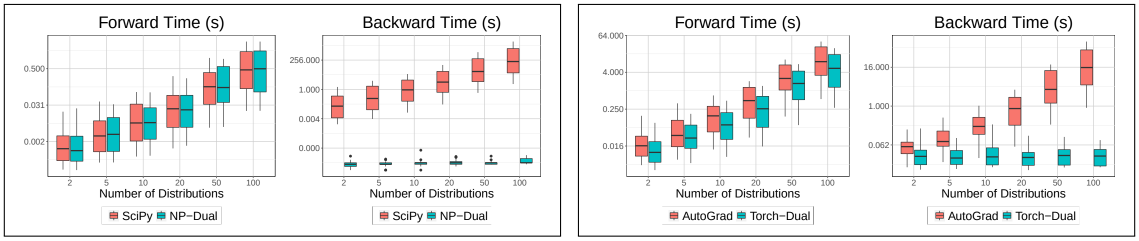


Figure 6.4: Forward/Backward Pass Times for 10 Bins with varying number of distributions, averaged over 10 runs. Forward wall-clock times are comparable, regardless of backend (*left pair*: NumPy+SciPy; *right pair*: PyTorch+AutoGrad). Direct reading of the gradient via the dual leads to significant gains in backward pass times, where automatic differentiation scales poorly with the number of distributions.

of distributions and bins increases, the time-cost for existing methods using more generic LP solvers can become significant, and may become infeasible with generic solutions via CVXPY (Diamond and Boyd, 2016). We compare our approach to two off-the-shelf implementations of barycenters via the Python Optimal Transport (POT) Library (Flamary et al., 2021), along with directly solving the Earth Mover's problem via CVXPY. Even for 10 distributions over 10 bins, CVXPY is unable to allocate the necessary memory using a direct instantiation.

Generalized EM Fairness on Fairness Datasets

With a viable tool in hand, we move to practical applications in machine learning fairness, which naturally requires enforcing closeness in model outputs. Here, we construct networks with our DEMD regularizer, where we discretize the final activation output prior to classification, and push the distributions of this activation to be similar among sensitive attributes.

Data. We identify 4 common fairness datasets often used to benchmark fair machine learning algorithms: (1) the German Credit Dataset (Hofmann, 1994), (2) the Adult Income Dataset (Dua and Graff, 2017), (3) the Communities and Crime Dataset (Redmond, 2009), and (4) The ACS Income dataset, recently made available as a large, population-level demographic dataset (Ding et al., 2021) containing as many as nine sensitive attributes. We set up a simple three-layer neural network for classification tasks with the addition of a fairness-type regularizer. We compare our construction with 4 off-the-shelf plug-in regularizers: (1) No regularization, (2) Demographic Parity (DP), (3) Equalized Odds (EO), and (4) a histogrammed barycenter construc-

Table 6.1: **Fairness Experiments.** Measures evaluated using standard metrics: maximum Demographic Parity Gap (**DP**), maximum Equalized Odds Gap (**EO**), and (**DEMD**). For all measures, lower values are preferred. With comparable accuracy, DEMD regularization leads to fairer representations as measured by common metrics. DP and EO measures are scaled by 100 for ease of presentation. Best results shown in bold.

	German			Adult			Crime			ACS-Income		
	DP	EO	DEMD	DP	EO	DEMD	DP	EO	DEMD	DP	EO	DEMD
None	$17_{(5)}$	$11_{(2)}$	$1.69_{(0.32)}$	$18_{(1)}$	$13_{(0)}$	$1.69_{(0.07)}$	$38_{(6)}$	$45_{(3)}$	$2.86_{(0.38)}$	$37_{(1)}$	$25_{(0)}$	$4.78_{(0.32)}$
DP-Reg.	$16_{(6)}$	$10_{(3)}$	$1.5_{(0.26)}$	$17_{(1)}$	$13_{(1)}$	$1.6_{(0.07)}$	$38_{(6)}$	$45_{(3)}$	$2.83_{(0.39)}$	$48_{(4)}$	$28_{(0)}$	$5.02_{(0.31)}$
EO-Reg.	$17_{(5)}$	$11_{(2)}$	$1.69_{(0.32)}$	$\mathbf{14}_{(1)}$	$\mathbf{12}_{(1)}$	$\mathbf{1.43}_{(0.07)}$	$38_{(5)}$	$\mathbf{44}_{(3)}$	$2.83_{(0.39)}$	$38_{(1)}$	$26_{(0)}$	$4.82_{(0.32)}$
Bary-POT	$27_{(5)}$	$17_{(1)}$	$1.5_{(0.21)}$	$18_{(1)}$	$13_{(0)}$	$1.64_{(0.07)}$	$\mathbf{36}_{(5)}$	$\mathbf{44}_{(4)}$	$2.81_{(0.3)}$	$57_{(37)}$	$50_{(44)}$	$4.38_{(0.16)}$
DEMD (ours)	$\mathbf{14}_{(7)}$	$\mathbf{9}_{(4)}$	$\mathbf{1.41}_{(0.35)}$	$15_{(1)}$	$\mathbf{12}_{(1)}$	$1.44_{(0.08)}$	$\mathbf{36}_{(6)}$	$\mathbf{44}_{(3)}$	$\mathbf{2.69}_{(0.44)}$	$\mathbf{33}_{(0)}$	$\mathbf{24}_{(0)}$	$\mathbf{3.6}_{(0.29)}$

tion. DP and EO regularizers were computed using a PyTorch version of FairLearn (Bird et al., 2020), and the barycenter version was implemented using POT library (with GPU backend). Because the scale of the regularization term is not directly comparable, we sweep regularization weights and select the best over all measures for a each dataset/regularizer pair. We use 10 bins and replicate all experiments over three random seeds.

Models. We set up two model settings with a standard logistic regressor and a 2-layer neural network. We compare three types of plug-in regularizers: (1) Demographic Parity (DP), (2) Equalized Odds (EO), and (3) the Generalized EMD. DP and EO regularizers were computed using a PyTorch implementation of FairLearn (Bird et al., 2020).

Results. In the summary in Table 6.1, models were selected with the largest regularization weight before accuracy dropped significantly. When accuracies are comparable, we see good performance (when minimizing DEMD regularizer) against baseline methods. Notably, when accuracies are similar across methods, minimizing DEMD tends to give better (lower) fairness measures across all datasets. **Summary:** DEMD on the final network layer helps control multiple fairness measures.

Table 6.2: **Harmonization Experiments.** Evaluations conducted along three metrics, Accuracy (**ACC**), Adversarial measure (**ADV**) and Maximum Mean Discrepancy (**MMD**). A lower value ↓ of ADV and MMD indicate successful harmonization across the different groups. A higher ACC with a small drop from baseline is preferred.

	German			Adult			Crime			ACS-Income		
	ACC ↑	ADV ↓	MMD ↓	ACC ↑	ADV ↓	MMD ↓	ACC ↑	ADV ↓	MMD ↓	ACC ↑	ADV ↓	MMD ↓
None	74(0.9)	93(1.3)	7.7(0.8)	84(0.1)	83(0.1)	9.8(0.3)	85(0.2)	77(0.5)	14(0.1)	78(0.1)	98(0.7)	160(2)
Li et al.	73(1.5)	92(1.3)	1.5(0.3)	84(0.1)	83(0.1)	3.1(0.3)	85(0.5)	76(1.6)	12(1.0)	78(0.1)	97(0.5)	17(1.2)
Xie et al.	76(1.3)	93(0.6)	1.2(0.2)	84(0.04)	81(0.7)	4.2(2.4)	85(0.2)	76(0.7)	15(0.6)	78(0.1)	94(5.6)	99(2.9)
DEMD	74(1.1)	93(0.4)	2.1(0.4)	84(0.03)	82(0.2)	5.3(1.4)	83(0.3)	72(1.0)	7.1(1.0)	77(0.4)	96(0.5)	26(3.9)

Harmonization for Invariant Representations

Having evaluated the use of the DEMD layer to constrain the neural network for fairness measures, we will now move to a more general problem of deriving invariant representations from the datasets. Here, invariance is sought in regard to the sensitive attributes. Recent works (Lokhande et al., 2022) on invariant representation learning propose leveraging an encoder-decoder architectures to map the dataset features to latent space representations. The latent space representations are penalized to match or harmonize the distributions across several groups in the dataset. In contrast to the previous section, the goal here is to identify a good mapping in the latent space that is devoid of any group-related information in the dataset. Consequently, our evaluation metrics test if the latent representations have a lower value of (i) ADV, adversarial measure and (ii) MMD, maximum mean discrepancy measure. The measure ADV tests to what extent can a separate neural network predict the group information from the latent features. Alternatively, the MMD scores measure the distance between the probability distributions across groups. Prior works such as Li et al. (2014) and Xie et al. (2017) optimize each of these measures separately. Our experiments (Table 6.2) show the DEMD layer performs competitively with the baselines when applied on the latent space. Interestingly, these performance gains come despite not directly optimizing the harmonization measures, in contrast to baselines, which require several practical adjustments (batch variants and secondary neural networks). **Summary**: DEMD as an intermediate layer can be used to derive invariant representations efficiently.

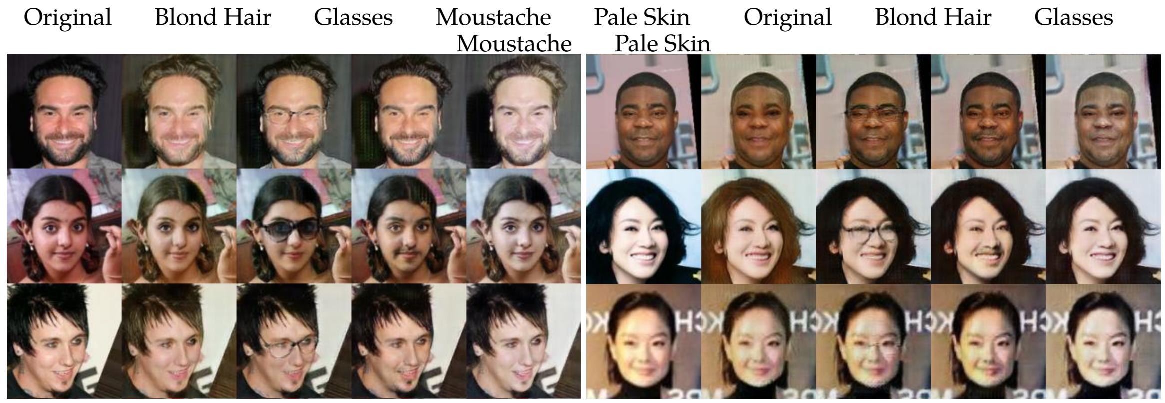


Figure 6.5: Six MWGAN+DEMD CelebA image translation results. Each row corresponds to a random sample in the validation set. The leftmost image is the original source image, and sequential columns represent translations to (1) Blonde Hair, (2) Glasses, (3) Moustache, and (4) Pale Skin. Using DEMD as a relaxation to the Multiple Marginal Matching problem facilitates high quality images translation. For more results see Appendix C.3.

Multi-domain Image Translation

We apply our construction to a recent multi-marginal GAN framing of multi-domain image translation. With the goal of learning a mapping for a source image to multiple target domains, a focal point of recent literature has been to reduce the computational requirements of training individual models for each target, and learn a concurrent matching problem over a number of shared parameters or networks. MWGAN (Cao et al., 2019) set up a Multiple Marginal Matching problem in this context. Extending upon their key observation that inter-domain constraints can be measured via the gradients for each domain, we instantiate our DEMD layer over the gradient norms computed for each sample in a batch per group. Specifically, DEMD minimizes the differences between the distributions over the gradient norms across all target domains. Using a weight of 100 for both regularizers, we observe similar performance when compared to the original MWGAN construction. In Figure 6.5 we show a few samples generated over an image translation task on the CelebA dataset. Here, we aim to translate original dataset images to ones with specific target attributes. **Summary:** Multi-domain image translation can be conducted by adding a DEMD layer in the native GAN construction.

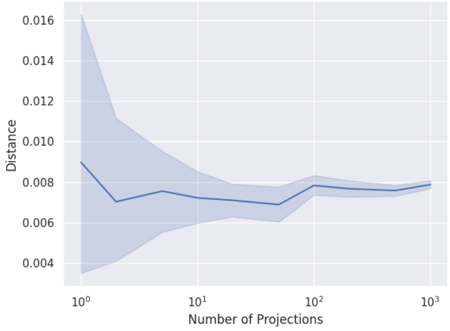


Figure 6.6: Sliced DEMD as a function of number of projections.

6.6 Conclusion

We presented an efficient solution for solving common practical multi-marginal optimal transport problems. Our construction is significantly cheaper to compute compared to similar methods, and allows for large numbers of distributions to be matched in common DNN pipelines. Our implementation allows imposing fairness constraints for a variety of applications, including those with many groups, without the need for pairwise measures. As such, subgroup fairness (Kearns et al., 2018) is an interesting problem setting that we believe can benefit: intersections of protected attributes exponentially increase the number of "subgroups", and as such an efficient method may be valuable. Other properties such as Minkowski additivity that have not been explicitly leveraged in our experiments may also be a worthwhile direction to explore.

One limitation of DEMD is its inability to be directly applied to distributions over multi-dimensional discrete spaces, such as latent spaces common in generative models. Slicing is a heuristic that has been shown to work well. To evaluate feasibility, we embed distributions over multi-dimensional continuous spaces, take random projections over 1-D spaces, and recompute our DEMD measure. Our gains in the many-distribution setting extend here as well: over a 64-dimensional latent space embedding of CelebA, we can efficiently compute our DEMD measure over all 40 attribute subgroups, and observe convergent behavior w.r.t. the number of projections.

Chapter 7

Follow-up and Future work

From features, parameters, and subsets, the above work has tackled the problem of modern subset selection uniquely informed by the particular method and application. Adapting ideas from scan statistics, differential geometry, tensor decompositions, conditional independence, and optimal transport, we are able to efficiently identify subsets important for myraid machine learning applications, including disease understanding, model compression, machine unlearning, and fairness. In the progression of this thesis, we have been able to concretely explore and contribute to the fast-growing developments in both theoretical machine learning as well as widespread application.

With all of this in place, we can now take a look at ongoing offshoots of this work building towards exciting and developing directions of research. This work has been and continues to be extended in a number of different theoretical and empirical directions, and here we briefly describe a few of them prior to a final note concluding the dissertation.

7.1 Analyzing Disease in Functional MRI via Covariance Trajectories

Work in Chapters 3 and 4 has been further developed for a number of other applications. The covariance trajectory analysis methods have been extended to the analysis of other Alzheimer's Disease populations (Hwang et al., 2020b), and work on understanding and localizing temporal lobe epilepsy measured over resting-state functional MRI acquisitions is under review.

Brain Network Abnormalities in Alzheimer's Disease

Rs-fMRI has been shown to be a valuable neuroimaging modality to study the pathophysiological mechanisms and effects of Alzheimer's Disease. However, most existing brain network modeling frameworks for rs-fMRI often do not account for the combined statistical and temporal dependencies underlying dynamic functional connectivity (dFC) in a statistically robust manner, which may be limiting our understanding of altered brain organization in disease. To address these issues, in this work we demonstrate that the covariance trajectory methods above can characterize dFC as covariance trajectories on the Riemannian manifold.

In this follow-up work, we leverage a different setup of the trajectory analysis in Chapter 3. Here, we have a much larger number of timepoints: compared to patient site visits, we have functional MRI measurements occurring at a rate of approximately once per second for scans ranging from 5 to 10 minutes. While the methods above can be applied, the computational cost, while not exponential, still grows at a rate infeasible for this application.

To address this, we construct a "windowed" approach, where we set specific window sizes and stride lengths such that our trajectory spans the length of the scan, and smoothly estimates the continuous trajectory through overlapping samples. We fit the trajectories for each subject and take corresponding means for the likelihood statistics and follow the hypothesis testing as above. Experimental results demonstrated that the approach is capable of identifying differential effects in large-scale functional networks altered in Alzheimer's Disease in a way that overcomes statistical challenges common with many neuroimaging studies.

Characterizing the Epilepsy Connectome

Building on the developments adapting these methods to functional MRI, we deploy and extend the windowed-approaches to the study of temporal-lobe epilepsy (TLE), where disease signals are known to cause widespread disruptions in connectivity dynamics within the well-studied epileptogenic network, but it is not yet clear if and to what extent other networks are perturbed in TLE.

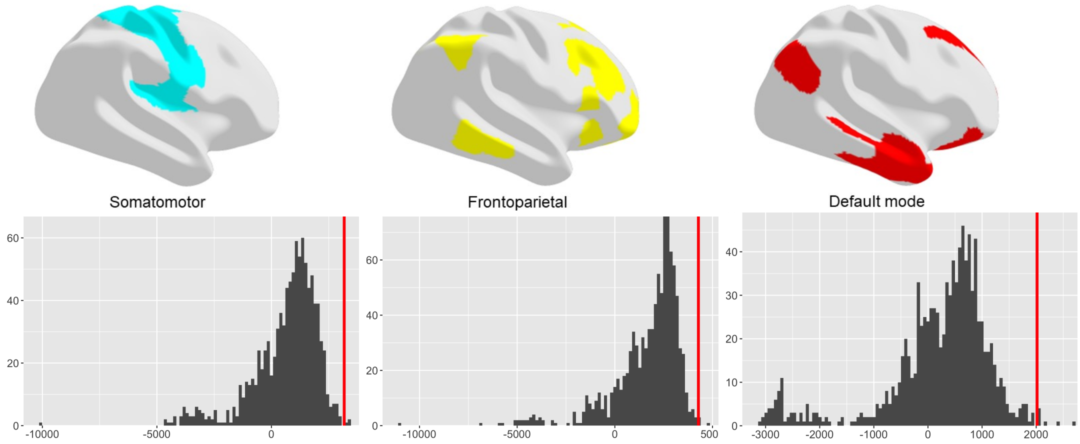


Figure 7.1: Functional networks exhibiting significant first- and second-order group differences with the corresponding estimated null distribution and alternative statistic (red) using our pipeline (brain visualization from right).

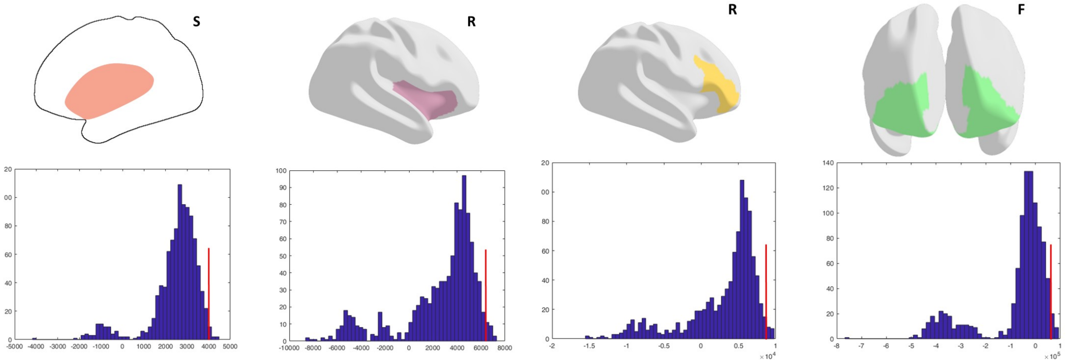


Figure 7.2: Networks exhibiting significant first- and second-order group differences using the proposed method with the corresponding estimated null distribution and alternative statistic: (from left to right) Subcortical, Insular and Frontal Opercular Cortex, Inferior Frontal Cortex, Orbital and Polar Frontal Cortex. S - saggital view, R - visualization from the right, F - visualization from the front.

Experimentally, we demonstrate that the approach identifies distinct subsets of temporally evolving connectivity features that stratify TLE patients and healthy controls, which map to altered connectivity at the network-scale.

The ideas in these chapters have also contributed to the understanding and modeling of uncertainty in recurrent models (Hwang et al., 2020a), and more generally towards efficient Bayesian methods for deep Monte Carlo methods (Nazarovs et al., 2021). Applications based on modeling disease progression as a function of both static and time-varying variables was extended to “deepify-ing" mixed effects models (Xiong et al., 2019b). Interesting developments on the tensor decomposition side have led to work combining ideas in earlier work inarchitecture search (Xiong et al., 2019c) with such compression methods, influencing high visibility publications by peers (Xiong et al., 2021).

7.2 Building Plug-and-Play Tools for Discrete Optimal Transport

The rate of development and research in machine learning and associated applications continues at its current pace largely due to easily deployable tools that operate seemlessly with existing workflows. PyTorch, Tensorflow, and all associated software packages have enabled this growth. In fact, concurrent to this work, a separate group had identified similar insights to ours, and was similarly reviewed and published with great interest at the same conference (Just et al., 2023). While there have been significant developments in Optimal Transport and its associated uses in deep learning, turnkey tools remain somewhat lacking for a number of reasons, as described in Chapter 6.

Building upon the theoretical efficiencies of the d-EMD solver and minimizer, we are developing a plug-in for the Python Optimal Transport library, with the goal of enabling optimal transport researchers to more easily develop methods for both pure transportation problems, as well as methods that incorporate with automatic differentiation pipelines. With the ability to solve larger-scale problems efficiently, users can more easily solve problems with high-dimensional data that would otherwise be infeasible with existing off-the-shelf solvers.

As noted, the efficient multi-marginal solution opens the door to a large number of domains unstudied due to computational cost. With approachable open-source code we hope to expand the tools described beyond one-dimensional distributions to problems in computer vision, both over two dimensional image spaces as well as beyond. New computational bottlenecks may arise as the distributions become higher-order tensors, which may themselves lead to the need for new technical insight.

Building upon the applications in fairness and the ideas related to subgroup fairness mentioned at the end of Chapter 6, we hope to specifically develop and merge existing fairness software stacks with our tools. Theoretically, subgroup fairness implies particular relationships among distributions, and as such interesting algorithmic and computational tools may be enabled under assumptions typical in subgroup settings.

7.3 Interpretability in Deep Models

Work in unlearning in Chapter 5 is relatively nascent, but excitement in the field is growing and a number of works have been motivated by recent successes. Methods for directly measuring, and perhaps even promoting or reducing conditional independence are beginning to be explored. Motivated by the discretization ideas in Chapter 6 alongside conditional independence, a particular direction of interest is that of general interpretability within continuous and typically black-box models. When a system is a large, complex function with high dimension and nonlinearity, expecting sparse solutions that *also* correspond to clear demarcations within the task or data is extremely optimistic.

Using methods described in the introduction such as activation maps or Shapley values typically leads to some form of “blame" in a continuous sense. These approaches are still being actively developed, and even now efficient approaches to “searching" or “identifying" over large supersets are being explored, with kernel and deep-learning based approaches in place to approximate these set functions.

These methods lead to a feature or parameter subset that has the largest (perhaps local) *influence*, but it unclear whether this translate directly to *dependence*. Conditional independence may be exactly the measure we are truly interested in, and equipped with a reasonable measure, it seems promising to explore its application within the broader space of interpretability. While the CODEC measure used here is somewhat simple, it does suffer from similar practical computational hurdles, albeit scaling polynomially compared to the direct set functions e.g., Shapley values. The extensions of CODEC as it serves as a drop-in comparison to existing measures looks to be an exciting direction of development as the interest and need for interpretability of large models grows.

A Final Note

As noted, current state-of-the-art machine learning research is moving at an accelerating pace. During the time of putting together this text, previously promising directions of research interest in the community have come and gone. The exceptional performance of transformer models has spurred exponential developments in language models, bleeding into vision domains where previous work primarily focused on convolutional methods.

The field is now at a point where almost all high-visibility work can only be done with large language models. The development and deployment of GPT-2, ChatGPT, and its contemporaries has all but enveloped interest in artificial intelligence and machine learning. While the ideas above could be independently researched, it is almost a requirement that they now be developed with application to large language models in mind.

Importantly, this accelerating pace of innovation does not appear to be slowing, and it is difficult to predict what will be the most important new development, given timelines for real-world performance beating state-of-the-art is now on the order of months.

The excitement of these developments notwithstanding, concerns surrounding ideas motivating this thesis are becoming more and more important as well as evident, with respect to fairness, accountability, and safety. Aligning models to the goals of stakeholders is of increasing importance as the models' ability to affect the world grows, and I hope that we as machine learning researchers, and as humans, rise to the task.

Appendix A

Second Order Group Differences Theoretical and Experimental Details

A.1 Technical Proofs.

Proof of Lemma 3.18

To remind the reader, this result was necessary in order to allow us to reduce the number of subgraphs (regions) that need to be evaluated over the graph. By bounding the covering number we have a guarantee that we do not need to consider an exponential number of subgraphs in order to find a localization.

Proof. To upper bound $N(A, \epsilon)$, we first construct the ϵ-covering set of $\mathcal{R}(A)$ under metric d. To this end, we decompose $\mathcal{R}(A)$ into several disjoint sets

$$\mathcal{R}_j(A) = \left\{ B(v, r) \in \mathcal{R}(A) : \left(1 - \frac{(j+1)\epsilon}{2}\right) A < |E(B(v, r))| \leq \left(1 - \frac{j\epsilon}{2}\right) A \right\},$$

for $j = 0, 1, \ldots, \lceil \frac{1}{\epsilon} \rceil$. Our strategy is to construct ϵ-covering set for each set $\mathcal{R}_j(A)$.

We only construct ϵ-covering set for $\mathcal{R}_0(A)$; $\mathcal{R}_j(A)$ $(j \geq 1)$ can be treated similarly. To construct the ϵ-covering set for $\mathcal{R}_0(A)$, we denote by $d_{v,r}$ the largest positive number such that

$$\frac{|E(B(v, r - d_{v,r}))|}{|E(B(v, r))|} \geq 1 - \frac{\epsilon}{2}, \tag{A.1}$$

for every $v \in V$ and $r \in \mathbb{N}$. Let $\mathcal{D}_1$ the collection of $d_{v,r}$ such that $B(v,r) \in \mathcal{R}_0(A)$, i.e.

$$\mathcal{D}_1 = \{d_{v,r} : B(v,r) \in \mathcal{R}_0(A)\},$$

and $\mathcal{V}_1$ the collection of nodes such that $B(v,r) \in \mathcal{R}_0(A)$, i.e.

$$\mathcal{V}_1 = \{v : B(v,r) \in \mathcal{R}_0(A)\}.$$

We pick up the largest number in $\mathcal{D}_1$, denoted by d_{v_1,r_1}, i.e. $d_{v_1,r_1} \geq d_{v,r} \; \forall \; d_{v,r} \in \mathcal{D}_1$ and define $\tilde{\mathcal{V}}_1$ as

$$\tilde{\mathcal{V}}_1 = \{v \in \mathcal{V}_1 : v \in B(v_1, d_{v_1,r_1}/2)\}.$$

After defining $\tilde{\mathcal{V}}_1$, $\mathcal{D}_2$ and $\mathcal{V}_2$ can be defined as

$$\mathcal{D}_2 = \mathcal{D}_1 \setminus \{d_{v,r} : v \in \tilde{\mathcal{V}}_1\} \qquad \text{and} \qquad \mathcal{V}_2 = \mathcal{V}_1 \setminus \tilde{\mathcal{V}}_1.$$

Then we can pick up the largest number in $\mathcal{D}_2$, denote by d_{v_2,r_2} and $\tilde{\mathcal{V}}_2$ can be defined similarly. We can repeat the above process until $\mathcal{D}_M$ and $\mathcal{V}_M$ are empty for some M.We actually obtain a partition of $\mathcal{V}_1$,

$$\bigcup_{i=1}^{M} \tilde{\mathcal{V}}_i = \mathcal{V}_1 \qquad \text{and} \qquad \tilde{\mathcal{V}}_{i_1} \cap \tilde{\mathcal{V}}_{i_2} = \emptyset \qquad 1 \leq i_1 < i_2 \leq M.$$

Based on $d_{v_1,r_1}, \ldots, d_{v_M,r_M}$, we are ready to prove the set

$$\mathcal{R}_0(A,\epsilon) = \{B(v_i,r_i) : 1 \leq i \leq M\}$$

is actually an ϵ-covering set for $\mathcal{R}_0(A)$. To this end, it is equivalent to show that for arbitrary $B(v',r') \in \mathcal{R}_0(A)$, we have

$$d(B(v',r'), B(v_i,r_i)) \leq \epsilon \tag{A.2}$$

when $v' \in \tilde{\mathcal{V}}_i$. To show (A.2), we consider two cases where $r' > r_i - d_{v_i,r_i}/2$ and $r' \leq r_i - d_{v_i,r_i}/2$. When $r' > r_i - d_{v_i,r_i}/2$, then

$$B(v_i, r_i - d_{v_i,r_i}) \subset B(v',r').$$

Combining above result, (A.1), and the definition of $\mathcal{R}_0(A)$ yields

$$\begin{aligned}&\frac{|E(B(v',r'))\cap E(B(v_i,r_i))|}{\sqrt{|E(B(v',r'))||E(B(v_i,r_i))|}}\\ \geq&\frac{|E(B(v_i,r_i-d_{v_i,r_i}))|}{\sqrt{|E(B(v',r'))||E(B(v_i,r_i))|}}\\ \geq&\sqrt{1-\frac{\epsilon}{2}}\frac{|E(B(v_i,r_i-d_{r_i}))|}{|E(B(v',r'))|}\\ \geq&1-\epsilon.\end{aligned}$$

On the other hand, if $r' \leq r_i - d_{v_i,r_i}/2$, then

$$B(v',r') \subset B(v_i,r_i). \tag{A.3}$$

By definition of $\mathcal{R}_0(A)$, we can get

$$\frac{|E(B(v',r'))\cap E(B(v_i,r_i))|}{\sqrt{|E(B(v',r'))||E(B(v_i,r_i))|}} \geq \sqrt{\frac{|}{E}(B(v',r'))||E(B(v_i,r_i))|} \geq 1-\epsilon.$$

Therefore, (A.2) is proved and $\mathcal{R}_0(A,\epsilon)$ is an ϵ-covering set for $\mathcal{R}_0(A)$.

The rest of the proof is to bound the cardinality of $\mathcal{R}_0(A,\epsilon)$, i.e. M. Note that (3.17) implies there exists some constant $D_{H,S}$ only depending on H and S such that, for any $v \in V$ and $r \in \mathbb{N}$,

$$|E(B(v,r/2))| \geq D_{H,S}|E(B(v,r))|.$$

By the definition of d_{v_i,r_i}, we can ensure $B(v_i, d_{v_i,r_i}/4)$ are disjoint. Hence, this implies

$$|E(\tilde{\mathcal{V}}_i)| \geq |E(B(v_i,d_{v_i,r_i}/4))| \geq D_{H,S}^2|E(B(v_i,d_{v_i,r_i}))| \geq D_{H,S}^2 HA\epsilon^S/2^{S+1}.$$

The last inequality is suggested by (3.17) and (A.1). The volume argument yields

$$M \leq \frac{|E|}{D_{H,S}^2 HA\epsilon^S/2^{S+1}} \leq \frac{2^{S+1}}{D_{H,S}^2 H}\frac{|E|}{A}\left(\frac{1}{\epsilon}\right)^S$$

(3.18) is obtained upon application of the above to each $\mathcal{R}_j(A)$. □

Proof of Theorem 3.19

Before we are ready to prove Theorem 3.19, we need the following result:

Lemma A.1. *Let $Y_1, \ldots, Y_d$ be i.i.d. standard Gaussian variable, i.e. $N(0,1)$ and $a_1, \ldots, a_d$ be a sequence of numbers. If*

$$Z = \sum_{i=1}^{d} a_i(Y_i^2 - 1), \tag{A.4}$$

then

$$\mathbb{P}(|Z| \geq 2|a|_2\sqrt{x} + 2|a|_\infty x) \leq 2\exp(-x) \tag{A.5}$$

where $|a|_2 = \sqrt{\sum_{i=1}^d a_i^2}$ and $|a|_\infty = \max_{i=1,\ldots,d} |a_i|$.

Proof. This is a direct extension of lemma 1 in Laurent and Massart (2000) to the negative case. We follow arguments similar to theirs. Let $\phi(x)$ be the the logarithm of the Laplace transform of $Y_i^2 - 1$. For any $-1/2 < x < 1/2$,

$$\phi(x) = \log\left(\mathbb{E}\left(\exp(x(Y_i^2 - 1))\right)\right) = -x - \frac{1}{2}\log(1 - 2x) \leq \frac{x^2}{1 - 2|x|}.$$

This leads to

$$\begin{aligned}\log(\mathbb{E}(e^{xZ})) &= \sum_{i=1}^{d} \log\left(\mathbb{E}\left(\exp(a_i x(Y_i^2 - 1))\right)\right) \\ &\leq \sum_{i=1}^{d} \frac{a_i^2 x^2}{1 - 2|a_i|x} \\ &\leq \frac{|a|_2^2 x^2}{1 - 2|a|_\infty x}\end{aligned}$$

With the same arguments in Laurent and Massart (2000), we could prove that

$$\mathbb{P}\left(Z \geq 2|a|_\infty x + 2|a|_2\sqrt{x}\right) \leq \exp(-x).$$

The other direction can be proved if we apply the same argument for $-Z$. □

With this in hand we proceed to prove Theorem 3.19.

Proof. In the following proof, C always refers to some constant, although its value may change from place to place. First, we prove (3.22). To this end, we prove concentration inequalities for L_R for some R and $L_{R_1} - L_{R_2}$ for some $R_1 \neq R_2$. Since we assume the noise follows normal distribution, we have

$$(\hat{\boldsymbol{\beta}}_1^R - \hat{\boldsymbol{\beta}}_2^R)^T \Sigma_R^{-1} (\hat{\boldsymbol{\beta}}_1^R - \hat{\boldsymbol{\beta}}_2^R) = \frac{\sum X_i^2 - (\sum X_i)^2}{2} \|\hat{\boldsymbol{\beta}}_1^R - \hat{\boldsymbol{\beta}}_2^R\|^2 \sim \chi^2_{|E(R)|}.$$

By tail bound for χ^2 random variables (see e.g. Laurent and Massart (2000)), we can yield

$$\mathbb{P}\left(L_R > 2t + \frac{2t^2}{\sqrt{|E(R)|}} \right) \leq \exp(-t^2). \tag{A.6}$$

By definition, $L_{R_1} - L_{R_2}$ can be written as

$$L_{R_1} - L_{R_2} = \frac{\sum_{i \in R_1 \setminus R_2} Z_i}{\sqrt{|E(R_1)|}} + \left(\frac{1}{\sqrt{|E(R_1)|}} - \frac{1}{\sqrt{|E(R_2)|}} \right) \sum_{i \in R_1 \cap R_2} Z_i - \frac{\sum_{i \in R_2 \setminus R_1} Z_i}{\sqrt{|E(R_2)|}}$$

where Z_i are independent random variable following distribution $\chi_1^2 - 1$. Lemma A.1 implies

$$\mathbb{P}\left(|L_{R_1} - L_{R_2}| > 2\sqrt{2d(R_1, R_2)}t + \frac{2t^2}{\min(|E(R_1)|, |E(R_2)|)} \right) \leq 2\exp(-t^2). \tag{A.7}$$

We now proceed to prove (3.22) by applying a chaining argument (See Talagrand (2006)) and concentration inequalities (A.6) and (A.7). Recall $\mathcal{R}_{app}(A, \epsilon)$ is the smallest ϵ-covering set of $\mathcal{R}(A)$ and $N(A, \epsilon)$ is the covering number of $\mathcal{R}(A)$. For any subgraph candidate R, we denote by

$$\pi_l(R) = \arg\min_{R' \in \mathcal{R}_{app}(A, e^{-l})} d(R, R').$$

For any $l^* > l_*$, which will be specified later, we write $\max_{R \in \mathcal{R}(A)} L_R$ into three parts

$$\max_{R \in \mathcal{R}(A)} L_R \leq \max_{R \in \mathcal{R}(A)} |L_R - L_{\pi_{l^*}(R)}| + \sum_{l=l_*}^{l^*-1} \max_{R \in \mathcal{R}(A)} |L_{\pi_{l+1}(R)} - L_{\pi_l(R)}| + \max_{R \in \mathcal{R}(A)} L_{\pi_{l_*}(R)}.$$

Now, we bound these three terms above separately.

Term 1. Let $l^* = 2\log|E|$. By concentration inequality (A.7) and union bound, we have

$$\begin{aligned}
&\mathbb{P}\left(\max_{R\in\mathcal{R}(A)} |L_R - L_{\pi_{l^*}(R)}| > \frac{2\sqrt{2(x+\log|E|)}}{|E|} + \frac{4x+8\log|E|}{A}\right)\\
\leq&|\mathcal{R}(A)|\mathbb{P}\left(|L_R - L_{\pi_{l^*}(R)}| > \frac{2\sqrt{2(x+\log|E|)}}{|E|} + \frac{4x+8\log|E|}{A}\right)\\
\leq&2\frac{|\mathcal{R}(A)|}{|E|^2}\exp(-x) \leq 2\exp(-x)
\end{aligned}$$

for $x < \log|E|$. Therefore, we have

$$\mathbb{P}\left(\max_{R\in\mathcal{R}(A)} |L_R - L_{\pi_{l^*}(R)}| > \frac{C(x+\log|E|)}{A}\right) \leq \exp(-x),$$

for $x < \log|E|$.

Term 2. Let $l_* = \log\log(|E|/A)$. Recall that the Avocado assumption (3.17) suggests that

$$N(A,\epsilon) \leq C_{H,S}\frac{|E|}{A}\left(\frac{1}{\epsilon}\right)^{S+1}. \tag{A.8}$$

Applying concentration inequality (A.6) along with

$$t = \sqrt{\log\left(\frac{|E|}{A}\right) + (S+1)\log\log\left(\frac{|E|}{A}\right) + x + C} \tag{A.9}$$

and the union bound, we have

$$\begin{aligned}
&\mathbb{P}\left(\max_{R\in\mathcal{R}(A)} L_{\pi_{l_*}(R)} > 2t + \frac{2t^2}{\sqrt{A}}\right)\\
\leq&N\left(A, \frac{1}{\log(|E|/A)}\right)\mathbb{P}\left(L_{\pi_{l_*}(R)} > 2t + \frac{2t^2}{\sqrt{A}}\right)\\
\leq&C_{H,S}\frac{|E|}{A}\left(\log\frac{|E|}{A}\right)^{S+1}\mathbb{P}\left(L_{\pi_{l_*}(R)} > 2t + \frac{2t^2}{\sqrt{A}}\right)\\
\leq&\exp(-x)
\end{aligned}$$

for $x < \log|E|$. Here we also apply condition (3.21). Therefore, we obtain

$$\mathbb{P}\left(\max_{R\in\mathcal{R}(A)} L_{\pi_{l_*}(R)} > 2\sqrt{\log\left(\frac{|E|}{A}\right) + (S+1)\log\log\left(\frac{|E|}{A}\right) + x + C}\right) \le \exp(-x)$$

for $x < \log|E|$.

Term 3. For any given l, application of concentration inequality (A.7), covering number condition (A.8), and the union bound yields,

$$\begin{aligned}
&\mathbb{P}\left(\max_{R\in\mathcal{R}(A)} |L_{\pi_{l+1}(R)} - L_{\pi_l(R)}| > \sqrt{\frac{C(\log(|E|/A)+l+x)}{e^l}} + \frac{C(\log(|E|/A)+l+x)}{A}\right)\\
\le & C_{H,S}\frac{|E|}{A}e^{(l+1)(S+1)}\\
&\mathbb{P}\left(|L_{\pi_{l+1}(R)} - L_{\pi_l(R)}| > \sqrt{\frac{C(\log(|E|/A)+l+x)}{e^l}} + \frac{C(\log(|E|/A)+l+x)}{A}\right)\\
\le & \frac{\exp(-x)}{l^2}.
\end{aligned}$$

for any $x < \log|E|$. With another standard application of the union bound, we have

$$\begin{aligned}
&\mathbb{P}\left(\sum_{l=l_*}^{l^*-1}\max_{R\in\mathcal{R}(A)} |L_{\pi_{l+1}(R)} - L_{\pi_l(R)}| > \sqrt{\frac{C(\log(|E|/A)+x)}{\log(|E|/A)}} + \frac{\log^2|E| + x\log|E|}{A}\right)\\
\le & \sum_{l=l_*}^{l^*-1}\mathbb{P}\left(\max_{R\in\mathcal{R}(A)} |L_{\pi_{l+1}(R)} - L_{\pi_l(R)}| > \sqrt{\frac{C(\log(|E|/A)+l+x)}{e^l}} + \frac{C(\log(|E|/A)+l+x)}{A}\right)\\
\le & \sum_{l=l_*}^{l^*-1}\frac{\exp(-x)}{l^2}\\
\le & 2\exp(-x).
\end{aligned}$$

Putting the three terms above together yields

$$\mathbb{P}\left(\max_{R\in\mathcal{R}(A)} L_R > 2\sqrt{\log\left(\frac{|E|}{A}\right)} + C(x+1)\right) \le \frac{4}{\log(e|E|/A)}\exp(-x),$$

where we apply $A \gg \log^2|E|$ and the inequalities $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$ and $\sqrt{a+b} \le \sqrt{a} + b/\sqrt{a}$.

Now, we apply this bound to $A = |E|2^{-k}$, $k \geq 0$ yielding

$$\mathbb{P}\left(\max_{R\in\mathcal{R}}\left(L_R - 2\sqrt{\log\frac{|E|}{|E(R)|}}\right) > C(x+1)\right) \leq 8\exp(-x).$$

This immediately suggests that $q_\alpha = O(1)$.

Now, let's turn to the case when a subgraph is significant, that is to prove (3.24). Assume the significant region is R_0. Using standard statistics we calculate the mean and variance of L_{R_0}

$$\mathbb{E}(L_{R_0}) = \frac{(\boldsymbol{\beta}_1^{R_0} - \boldsymbol{\beta}_2^{R_0})^T\Sigma_{R_0}^{-1}(\boldsymbol{\beta}_1^{R_0} - \boldsymbol{\beta}_2^{R_0})}{\sqrt{|E(R_0)|}}$$

$$Var(L_{R_0}) = 2 + 4\frac{(\boldsymbol{\beta}_1^{R_0} - \boldsymbol{\beta}_2^{R_0})^T\Sigma_{R_0}^{-1}(\boldsymbol{\beta}_1^{R_0} - \boldsymbol{\beta}_2^{R_0})}{|E(R_0)|}.$$

By Chebyshev's inequality, we have

$$\mathbb{P}\left(\frac{|L_{R_0} - \mathbb{E}(L_{R_0})|}{\sqrt{Var(L_{R_0})}} > x\right) \leq \frac{1}{x^2}. \tag{A.10}$$

If $(\boldsymbol{\beta}_1^{R_0} - \boldsymbol{\beta}_2^{R_0})^T\Sigma_{R_0}^{-1}(\boldsymbol{\beta}_1^{R_0} - \boldsymbol{\beta}_2^{R_0}) \geq |E(R_0)|$, then (A.10) suggests

$$\mathbb{P}(L_{R_0} > \sqrt{|E(R_0)|}) \to 1, \ |E| \to \infty$$

by taking x as a sequence (e.g., $\log\log(|E(R_0)|)$) which increases slow enough in (A.10). This leads to (3.24). If

$$(\boldsymbol{\beta}_1^{R_0} - \boldsymbol{\beta}_2^{R_0})^T\Sigma_{R_0}^{-1}(\boldsymbol{\beta}_1^{R_0} - \boldsymbol{\beta}_2^{R_0}) < |E(R_0)|,$$

then $Var(L_{R_0}) < 6$ and so (3.23) and (A.10) imply

$$\mathbb{P}\left(L_{R_0} - 2\sqrt{\frac{|E|}{|E(R_0)|}} > q_\alpha\right) \to 1, \qquad \text{as } |E| \to \infty.$$

□

A.2 Implementation Details.

The workflow below describes one run of our model given a sparsity is specified for the oracle graph procedure.

1. **Oracle Graph.** As noted in the main thesis, we use *graphical lasso* (*glasso*) to generate an *oracle graph*, which allows to define structured regions (subgraphs) for scan statistics on graphs. Each element of the input matrix C in (2.13) for glasso is generated by calculating the slope for each position of the covariance matrix across the predictors for each group, and then taking the difference between the groups. Equation (2.13), repeated here, is then solved using existing MATLAB interfaces to fast C implementations.

   $$\Theta = \arg\min_{\Theta \succeq 0} \quad -\log|\Theta| + tr(C\Theta) + \lambda||\Theta||_1 \tag{A.11}$$

   With sparsity parameter λ, this procedure generates a reasonably sparse *oracle graph*.

2. **Candidate Subgraphs.** With the oracle graph in hand, we then construct the set of all ball subgraphs, as defined in Section 3.4. By limiting ourselves to only a few $(D|V|)$ subgraphs, we can perform scan statistics more efficiently.

3. **Characterizing the Null Distribution.** In the case where we have few samples, we cannot directly apply the χ^2 result. In these cases, the null distribution is then characterized using permutation testing over all candidate subgraphs. For each subgraph the input data is permuted a number of times to generate a good representation of the distribution at that subgraph. All normalized (but not size-corrected) scan statistics are then calculated for all permutations across all subsets and then combined in order to create the null distribution.

4. **Calculating the Test Statistic** For a specific subset of the data, the scan statistic is calculated and corrected as described in Section 3.4, over the original grouping of the data. For each group, the logitudinal-covariance GLM (3.7) is computed using the procedures in Section 3.2.

5. **Region Identification.** We first identify all subsets whose statistic falls above the α-level threshold specified. Then the subset-collection procedure outlined in the

main thesis, developed by Jeng et al. (2010), is applied, and the non-overlapping critical regions are output.

Numerical Considerations

In practice, our empirical covariance matrices calculated on the sample data may not be positive definite. The matrix can be rank deficient when we do not have enough linearly independent samples. In addition, we may use a rank correlation matrix in its place, which also may not be PD. To resolve this issue, we *project* the empirical covariance matrix onto the symmetric-positive definite $\mathrm{SPD}(n)$ manifold. We first apply a standard procedure for transforming a symmetric matrix into a symmetric positive semidefinite (SPSD) one. As described in Wu et al. (2005), the standard eigenvalue thresholding, or clipping, $\lambda_{SPSD} = \max(0, \lambda)$ is sensible since it provides the optimal projection of any matrix onto the SPSD manifold. Let $\Sigma = U\Lambda U^{\top}$ be the eigenvalue decomposition of the matrix Σ. The SPSD projection of Σ is then $\text{proj}_{SPSD}(\Sigma) = U\text{diag}(\max(\lambda_1, 0), \ldots, \max(\lambda_n, 0))U^{\top}$. And so to project to the $\mathrm{SPD}(n)$ manifold we can simply add some epsilon to each element of the diagonal:

$$\text{proj}_{SPD}(\Sigma) = U\text{diag}(\max(\lambda_1, 0), \ldots, \max(\lambda_n, 0))U^{\top} + \epsilon I \tag{A.12}$$

A remark on the term ϵI will be useful here. We find that in experiments, numerical problems can arise if the smallest eigenvalue of the projected matrix is too small. By iteratively adding a small ϵ until the smallest eigenvalue is above our threshold, we ensure that the matrix is positive definite for the exponential and logarithmic maps. They are necessary for moving back and forth between the manifold and the tangent space.

A note on localization accuracy

In addition to simply checking whether or not we were able to correctly answer the hypothesis test group difference, it is important that if a significance is found, that it is found in the features that were originally used to generate the data. Using the same simulation setup as previous, we take the union of all subsets returned to be significant and check if each of the truly changing features p_t are contained within the superset.

In this particular case we find that our localization is only dependent on the graphical lasso procedure we use to generate the oracle graph. As long as the sparsity specified is large enough to include at least p_t edges, we find that in *every* simulation where we find a significant difference, the features that express the difference are a superset of the true features.

A.3 Preclinical AD Extended Details and Results.

Data and Variable Descriptions

In our neuroimaging experiments, a large number of our features describe specific and localized regions of the brain across multiple imaging modalities. Below we list and describe each of regions for each modality, and give a brief background on each of methods used to acquire the data. We also include the list of cognitive scores used in our analysis.

PET Imaging

Positron emission tomography has become an increasingly popular method of imaging the brain, specifically in the areas where cognitive decline can be strongly correlated with the specific matter being imaged. Pittsburgh compound B (PiB) was used as the tracer for these images, and the 16 mirrored (Left and Right) regions labeled below were selected as strongly correlated with the development and progression of Alzheimer's Disease.

1. PiB Angular L/R
2. PiB Cingulum Ant L/R
3. PiB Cingulum Post L/R
4. PiB Frontal Med Orb L/R
5. PiB Precuneus L/R
6. PiB SupraMarginal L/R
7. PiB Temporal Mid L/R
8. PiB Temporal Sup L/R

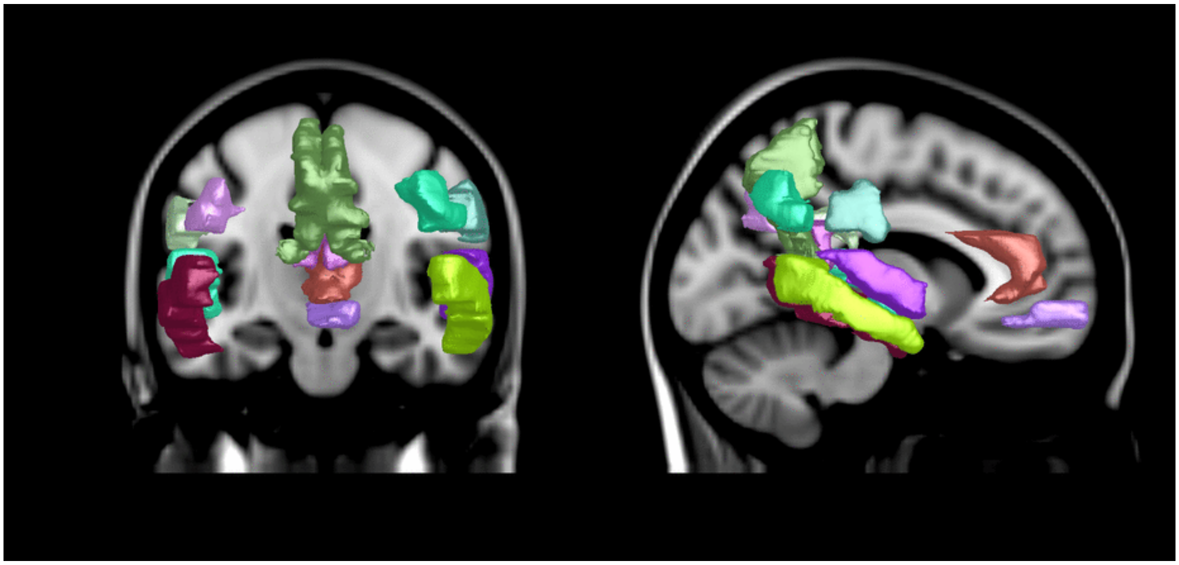


Figure A.1: 16 Positron Emission Tomography (PET) regions.

The average of the voxel values in each ROI (region of interest) of the brain are used for imaging features. The 16 regions are highlighted in Figure A.1.

DTI Imaging

Diffusion tensor imaging is used to measure the restricted diffusion of water through and about regions of the brain. The 48 regions here are the aggregated measurements of total rates of diffusion for each voxel in that region. The two measurements, Fractional Anisotropy (FA) and Mean Diffusivity (MD) collectively well describe the diffusion in a specific region. The following is the full list of regions used in our analysis. Regions that spanned across both the left and right sides of the brain are indicated as such, and were treated as separate and independent in our analyses.

1. Middle cerebellar peduncle
2. Pontine crossing tract (a part of MCP)
3. Genu of corpus callosum
4. Body of corpus callosum
5. Splenium of corpus callosum
6. Fornix (column and body of fornix)
7. Corticospinal tract R/L
8. Medial lemniscus R/L
9. Inferior cerebellar peduncle R/L
10. Superior cerebellar peduncle R/L
11. Cerebral peduncle R/L
12. Anterior limb of internal capsule R/L
13. Posterior limb of internal capsule R/L
14. Retrolenticular part of internal capsule R/L
15. Anterior corona radiata R/L
16. Superior corona radiata R/L
17. Posterior corona radiata R/L
18. Posterior thalamic radiation (include optic radiation) R/L
19. Sagittal stratum (include inferior longitidinal fasciculus and inferior fronto-occipital fasciculus) R/L
20. External capsule R/L
21. Cingulum (cingulate gyrus) R/L
22. Cingulum (hippocampus) R/L
23. Fornix (cres) / Stria terminalis (can not be resolved with current resolution) R/L
24. Superior longitudinal fasciculus R/L
25. Superior fronto-occipital fasciculus (could be a part of anterior internal capsule) R/L
26. Uncinate fasciculus R/L
27. Tapetum R/L

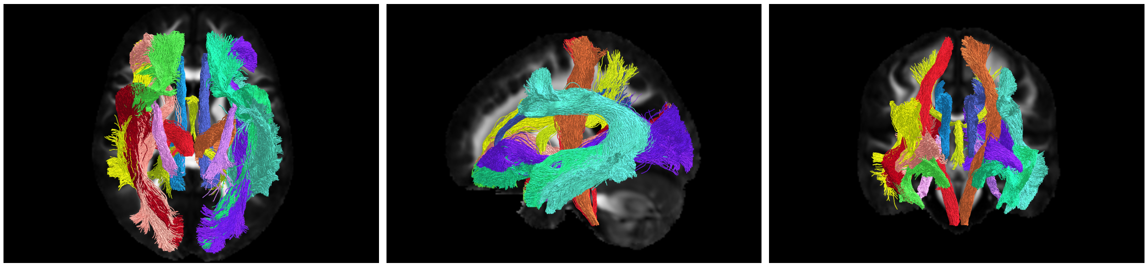


Figure A.2: 17 major DTI fiber bundles measured using Fractional Anisotropy (FA). The 48 selected for our analysis include a subset of these, which have been identified as critical regions that signal the beginnings of cognitive impairment.

Cognitive Evaluations

The battery of cognitive test scores in our analysis included a breadth of evaluations chosen specifically for their coverage of various measures of cognition. Among all tests given to the cohort, the following 17 were selected by expert clinicians and researchers in the field for their coverage and their potential value in understanding trends across groups.

1. WAIS-III Digit Span Forward Raw Score
2. WAIS-III Digit Span Backward Raw Score
3. WAIS-III Letter-Number Sequencing Raw Score
4. COWAT CFL Score
5. Boston Naming Test Total Score
6. RAVLT Learning Trial A1 Raw Score
7. RAVLT Learning Trial A2 Raw Score
8. RAVLT Learning Trial A3 Raw Score
9. RAVLT Learning Trial A4 Raw Score
10. RAVLT Learning Trial A5 Raw Score
11. RAVLT Learning Trial A6 Raw Score
12. RAVLT Delayed Recall Raw Score
13. Stroop Word/Color-Word Scaled Score
14. Trail-Making Test Part A
15. Trail-Making Test Part B
16. Clock Drawing Test Score
17. Center for Epidemiologic Studies Depression Scale Score

WAIS-III. This is the most widely used IQ test. The Digit Span examination is specifically meant to evaluate the working memory of an individual. Participants are required to attempt to recall a series of numbers in order, both forwards and backwards. Letter-Number sequencing reflects a similar idea, but with a mix of both numbers and letters in increasing and alphabetical order, and is meant to be an indicator of more complex mental control Wechsler (2014).

Rey Auditory Visual Learning Test. This test is specifically meant to evaluate all aspects of memory. Each trial evaluates a different type of memory, ranging from short-term and working memory to procedural and episodic memory. Schmidt et al. (1996).

Trail-Making Test. This is a very popular test in providing information about executive function in the brain. The test consists of drawing lines among a randomly generated set of points in a square, where each point is labeled with a number. In Part A, participants must 'connect the dots' in increasing numerical order, and in Part B in increasing numerical and alphabetical order. The score on the test is primarily dictated by the time in seconds it takes to complete the task for 25 of these 'dots.' More background information and normative analyses can be found in Tombaugh (2004).

Other tests similarly measure various cognitive function. While the Depression Scale Score did not crop up in any of our analyses here, it has been shown that depression is strongly associated with AD-related decline Wragg and Jeste (1989).

Detailed Imaging with Cognitive Tests Results

In the following tables we provide additional details of the statistical test we performed on the preclinical AD cohort. Each set contains a set of features found to display significant group difference (at the $p \leq 0.05$ level) along the covariance trajectory divided by the group variable indicated.

While some of these associations are well-known, few have been indicated as novel by AD researchers and clinicians, and to be of interesting value for further analysis.

Detailed results on larger cohort with only cognitive scores

We also applied our method to a larger cohort consisting of approximately 1500 subjects with varying temporal measurements on the battery of cognitive tests. Each individual had approximately 3 visits worth of data, and so our total number of

Amyloid Load (PiB Positivity)		
Set 1	PiB Angular L/R	PiB Cingulum Ant L/R
	PiB Cingulum Post L/R	PiB Frontal Med Orb L/R
	PiB Precuneus L/R	PiB Temporal Sup L/R
	PiB Temporal Mid L/R	**PiB SupraMarginal L**
Set 2	FA Cerebral peduncle R	FA Cerebral peduncle L
	MD Corticospinal tract R	MD Corticospinal tract L
	Trail-Making Test Part A Score	MD Cerebral peduncle R
	PET Cingulum Post R	

Table A.1: Group difference across Amyloid Load (PiB Positivity)

Gender		
Set 1	Rey Audio and Verbal Learning Test	FA Cingulum L
Set 2	FA Medial lemniscus L	FA Cingulum (hippocampus) L
	FA Posterior thalamic radiation (include optic radiation) L	
Set 3	FA Corticospinal tract R	FA Superior fronto-occipital fasciculus R

Table A.2: Group difference in gender

Genotype: APOE4		
Set 1	Digit Span Backward Raw Score	Stroop Color-word
	PiB Cingulum Post L	PiB Cingulum Post R
	PiB Frontal Med Orb L	PiB Frontal Med Orb R
	PiB Precuneus L	PiB Precuneus R
	PiB SupraMarginal	PiB Temporal Mid R

Table A.3: Group difference across Genotype APOE4 expression

Consensus Conference		
Set 2	Digit Span Backward Raw Score	Stroop Color-word
	PiB Cingulum Post L	PiB Cingulum Post R
	PiB Frontal Med Orb L	PiB Frontal Med Orb R
	PiB Precuneus L	PiB Precuneus R
	PiB SupraMarginal	PiB Temporal Mid R

Table A.4: Group difference across Expert MCI Diagnosis

Algorithmic Cognitive Impairment		
Set 1	Boston Naming Test Total Score	RAVLT Learning Trial A1 Raw Score
	RAVLT Learning Trial A6 Raw Score	

Table A.5: Group Difference Localization Across Algorithmic Impairment

measurements was approximately $n = 4000$. In addition to the groupings used above, we were able to use an algorithmic cognitive impairment (ACI) measure to further evaluate the model against a factor which is known to be group-separating. Below are the tabulated feature sets identified by our model for each of the group separations described in the main thesis. In this case to increase interpretability of the results we limited our search to groups of 3-6 features.

When grouped by genotype, the most indicative subset as shown in Table A.6. These tests are most closely associated with memory, and we see that no tests of executive function or spatial ability (Trail-Making or Clock Drawing) were included.

In addition to an algorithmic measure of impairment, a conference of expert clinicians and researchers have given each individual a clinical impairment diagnosis for each time they underwent the cognitive battery. Using this as a group separator, we found a large number of overlapping subsets that displayed significant group difference at the $p = 0.05$ level. These are shown in Table A.7. Trail-Making Test Parts A and B appeared in all identified subsets.

Genotype: ApoE4		
Set 1	WAIS-III Digit Span Backward Raw Score	RAVLT Learning Trial A3 Raw Score
	RAVLT Learning Trial A4 Raw Score	RAVLT Learning Trial A5 Raw Score

Table A.6: Group Difference Localization Across ApoE4 Genotype

Expert Consensus Measure	
WAIS-3 Letter-Number Sequencing Raw Score	Boston Naming Test Total Score
RAVLT Learning Trial A2 Raw Score	RAVLT Learning Trial A3 Raw Score
RAVLT Learning Trial A4 Raw Score	RAVLT Learning Trial A5 Raw Score
RAVLT Learning Trial A6 Raw Score	RAVLT Delayed Recall Raw Score
Trail-Making Test Part A	Trail-Making Test Part B
Clock Drawing Test Score	Center for Epidemiologic Studies Depression Scale Score

Table A.7: Group Difference Localization Across Expert Clinical Diagnosis

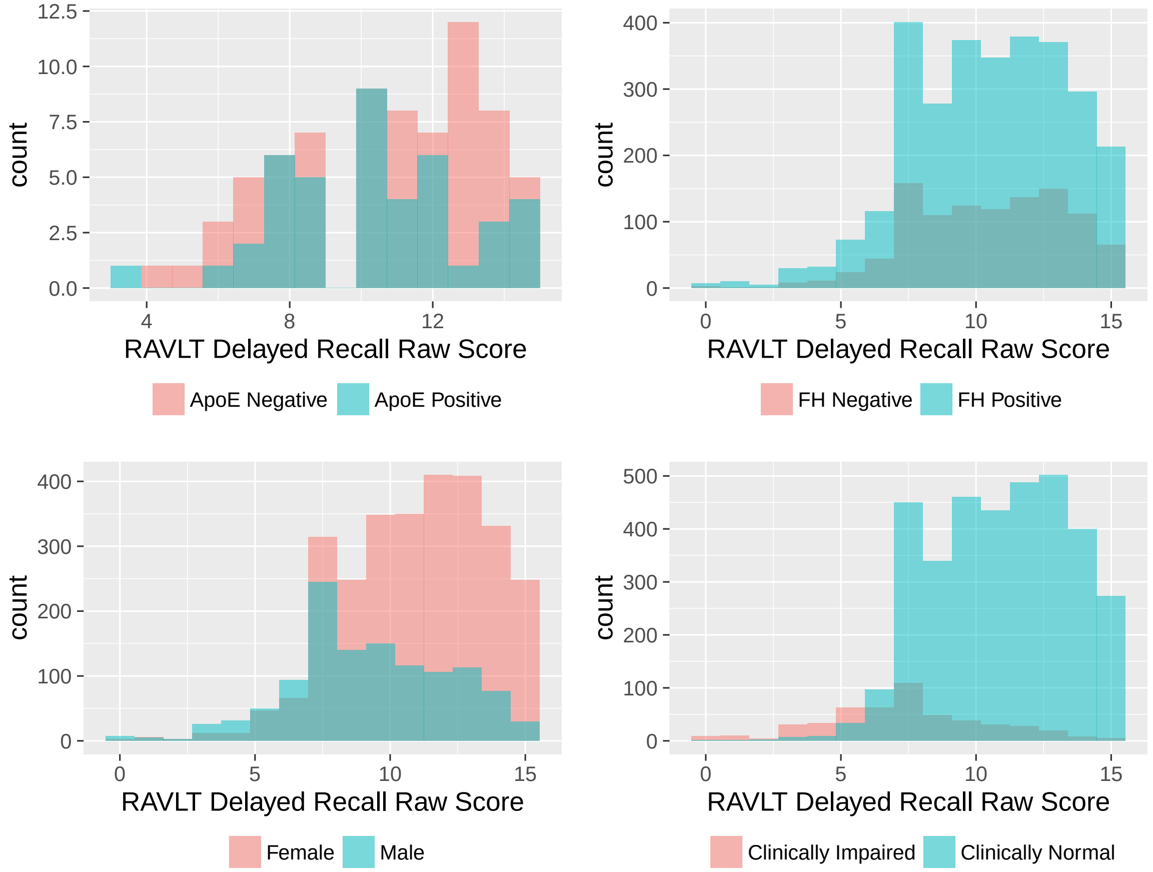


Figure A.3: Histograms of the Delayed Recall Scores for all time points for the ~ 4000 individual measurements across different group separations. We note in particular that the results found from the genotype separation above would have been hard to identify since given the distributions are extremely overlapping (top left) for this particular separation.

Appendix B

Conditional Independence and Unlearning Theoretical and Experimental Details

B.1 Theoretical Results

Proof of Lemma 1

Let us take D to be the training set; w.l.o.g. z is the point being removed. Let the residual dataset be $D' = D \setminus z$. Denote w^{-}_{Full} as the weight parameters after doing a full Hessian update and w^{-}_{Foci} as the weight parameters after doing a FOCI selected Hessian update. In an ideal case, we want $(w^{-}_{Foci}, D')/(w^{-}_{Full}, D')$ to be as close as possible to (w^{*}, D'). Note that we consider both (w^{*}, D) and (w^{*}, D') to be 0 as we don't expect model parameters to change drastically for one sample once trained to convergence.

Lemma B.1. *The gap between the gradient residual norm of the FOCI Unlearning update in Algorithm 4 1 and a full unlearning update via Eq. (5.4) in the main thesis,*

$$||\nabla\mathcal{L}(w^{-}_{Foci}, D')||_2 - ||\nabla\mathcal{L}(w^{-}_{Full}, D')||_2 \tag{B.1}$$

shrinks as $O(1/n^2)$.

Proof. Let w to be a network of many linear layers with possible activation functions; we can think of the norm as the sum of norm of gradients for each layer. Hence, for any model parameters w and dataset D, we have:

$$||\nabla\mathcal{L}(w,D)||_2 \coloneqq \sum_{l\in L} ||\nabla\mathcal{L}(w_l,D)||_2 \tag{B.2}$$

FOCI identifies a subset $T \subset L$ slices or layers that are to be updated. Let $R = L\backslash T$ be the remainder of the network which is not updated. Hence, B.2 for (w^{-}_{Foci}, D') can be written as:

$$||\nabla\mathcal{L}(w^{-}_{Foci},D')||_2 \coloneqq \sum_{l\in L} ||\nabla\mathcal{L}(w^{-}_{Foci_l},D')||_2 \tag{B.3}$$

$$= \sum_{l\in T} ||\nabla\mathcal{L}(w^{-}_{Foci_l},D')||_2 + \sum_{l\in R} ||\nabla\mathcal{L}(w^{-}_{Foci_l},D')||_2 \tag{B.4}$$

$$= \sum_{l\in T} ||\nabla\mathcal{L}(w^{-}_{Foci_l},D')||_2 + \sum_{l\in R} ||\nabla\mathcal{L}(w^{*}_{l},D')||_2 \tag{B.5}$$

The last line follows from the fact that layers in R are not updated.

We will next show how for the remainder of the dataset D', the changes in T propagate minimally when there are a large number of data points, n in the training set.

W.L.O.G. assume that we have a 3 layer network with the form:

$$(L_3(L_2(L_1(x)))) \tag{B.6}$$

For the point being removed $z := (x, y)$; let L_2 be the intermediate layer which is selected for update by FOCI. Before the update, activations out of L_2 are of the form $a_2 = L_2(L_1(x)) = L_2(a_1)$. After the update, activations out of L_2 can be written as:

$$a'_2 = L'_2(L_1(x)) = L'_2(a_1) \tag{B.7}$$

$$= w'_2 a_1 \tag{B.8}$$

$$= (w_2 + \delta_{w_2})a_1 \tag{B.9}$$

$$= w_2 a_1 + \delta_{w_2} a_1 \tag{B.10}$$

$$= a_2 + \delta_{w_2} a_1 \tag{B.11}$$

The Second line follows because L_1 isn't updated. For the following layer L_3, we have

$a_3 = L_3(a_2)$ before the update. After,

$$\begin{aligned}
a_3' &= L_3(a_2') \quad &\text{(B.12)}\\
&= L_3(a_2 + \delta_{w_2} a_1) \quad &\text{(B.13)}\\
&= L_3(a_2) + \nabla L_3(a_2)\delta_{w_2} a_1 + \mathcal{O}((\delta_{w_2} a_1)^2) \quad &\text{(B.14)}\\
&= L_3(a_2) + 0 + \mathcal{O}((\delta_{w_2} a_1)^2) \quad &\text{(B.15)}
\end{aligned}$$

The first-order term goes to zero, as L_3 has not been updated and we assume full model convergence.

For the Sekhari et al. (2021) update.

$$\delta_{w_2} = \frac{1}{(n-1)}(\hat{H}^{-1}) \sum_{z \in \{(x_k, y_k)\}} \nabla f(\hat{w}, z) \tag{B.16}$$

Hence, $\delta_{w_2}^2 \propto \frac{1}{n^2}$. Therefore, for large values of n, the third term in the equation above approaches 0. So, $a_3' = L_3(a_2)$. This shows that propagation is minimal. Similar arguments regarding null space for over-parameterized deep networks have been mentioned in Golatkar et al. (2020b).

Now, looking back at the residual gradient norm, we have:

$$\begin{aligned}
||\nabla \mathcal{L}(w^{-}_{Foci}, D')||_2 = \sum_{l \in T} ||\nabla \mathcal{L}(w^{-}_{Foci_l}, D')||_2 + \quad &\text{(B.17)}\\
\sum_{l \in R} ||\nabla \mathcal{L}(w_l^*, D')||_2 \quad &\text{(B.18)}
\end{aligned}$$

Based on the above argument of minimal propagation, the second term above goes to 0 for layers/slices in R. Therefore,

$$||\nabla \mathcal{L}(w^{-}_{Foci}, D')||_2 = \sum_{l \in T} ||\nabla \mathcal{L}(w_l^{-}, D')||_2 \tag{B.19}$$

and as such the gap between this and the full update is only the difference on the set R, shrinking as $O(1/n^2)$. □

Proof of Theorem 5.6

Theorem B.2. *Assume that layer-wise sampling probabilities are nonzero. Given (user specified) unlearning parameters ϵ, δ, the unlearning procedure in Algorithm 4 is (ϵ', δ')–forgetting* *where $\epsilon' > \epsilon, \delta' > \delta$ represent an arbitrary precision (hyperparameter) required for unlearning. Moreover, iteratively applying our algorithm converges exponentially fast (in expectation) with respect to the precision gap, that is, takes (at most) $O(\log \frac{1}{\mathbf{g}_\epsilon} \log \frac{1}{\mathbf{g}_\delta})$ iterations to output such a solution where $\mathbf{g}_\epsilon = \epsilon' - \epsilon > 0, \mathbf{g}_\delta = \delta' - \delta > 0$ are gap parameters.*

Proof. Our proof strategy is to show that our update step in Algorithm 1 is a specific form of Randomized Block Coordinate Descent (R-BCD) method. Then, we simply apply existing convergence rates of RBCD for general smooth minimization problems. In particular, our method can be seen as an extension of SEGA method in Corallary A.7. Gorbunov et al. (2020) where the descent direction is provided by using inexact inverse hessian metric Loizou and Richtárik (2020). The key difference in our setup is that the sampling probabilities are computed using the CODEC procedure instead of the random sampling at each step. We make the following three observation in our setup that immediately asserts correctness of the procedure.

First, by our construction in equation (5.11) in the main thesis, the sampling probabilities have full support. That is, the probability of selecting a particular weight in the neural network is strictly positive since $\xi \sim \mathcal{N}(0, \sigma^2), \sigma > 0$ is a continuous distribution which has unbounded support. Second, the overall rate of speed of convergence depends on the condition number of the (fixed) Hessian at the optimal solution since exact (ϵ, δ) unlearning is equivalent to linear least squares problem. Third, our update step is equivalent to a projected (or sketched) primal step, see equation 13 in (ArXiv Version Loizou and Richtárik (2019)). From these observations, we can see that our overall method is equivalent to SEGA in Gorbunov et al. (2020) or its noisy extension since we use only a small set of samples (to be unlearned) at each iteration. Consequently, we obtain the deterministic geometric rate of convergence (in expectation) by applying Corollary A.8. where σ in their work corresponds to the $\epsilon' - \epsilon > 0$ gap in our setup. Now, to get the probabilistic ϵ', δ' unlearning guarantee for the solution presented by our algorithm, we use Lemma 10 in Sekhari et al. (2021) on the solution returned, completing our proof. □

B.2 Experimental Details

Experiments were conducted using PyTorch 1.8 and CUDA Toolkit 10.2, run on Nvidia 2080 TIs and individual Nvidia A100s. Parallelization only occurred across runs;

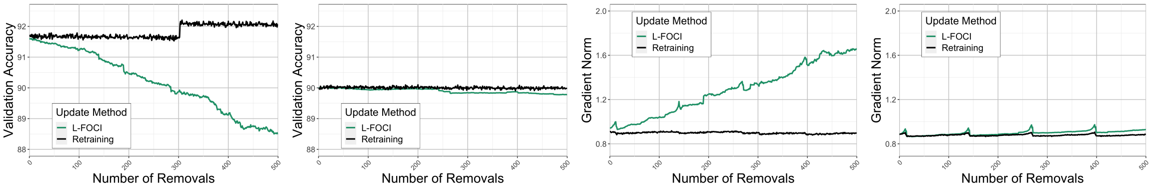


Figure B.1: MNIST Retraining results, comparing the effect of weight decay on unlearning via our LFOCI unlearning scheme and retraining.

the attached code can be run with any CUDA/PyTorch setup with the appropriate dependencies.

Markov Blanket Selection

Experimental settings were taken from Yang et al. (2020), with code adapted from `https://github.com/syanga/model-augmented-mutual-information`. 5000 samples were used for generating the data, and 100 trials/perumutations were conducted for the CIT testing framework.

MNIST Toy Results

Training for MNIST Logistic Regressor models was run using SGD with a learning rate of 0.1, batch size of 256, and weight decay of 0.01 for 50 epochs. 1000 perturbations were used for distribution approximation. Privacy parameters were set to $\epsilon = 0.1, \delta = 0.01$. Figures and numbers in the main thesis were averaged over 10 replications, for a random choice of 1000 samples to scrub.

Retraining Comparisons

MNIST: Affects of l_2 Regularization and Weight Decay

Repeating the retraining comparison in the main thesis with a larger regularization, we see that the effects of removal are significantly diminished and the model can support a larger number of removals before large performance drops.

CIFAR Retraining Comparisons: A Note on Batch Normalization

An important requirement for our retraining experiment is that our residual training set used for both scrubbing validation and retraining is able to take on any size,

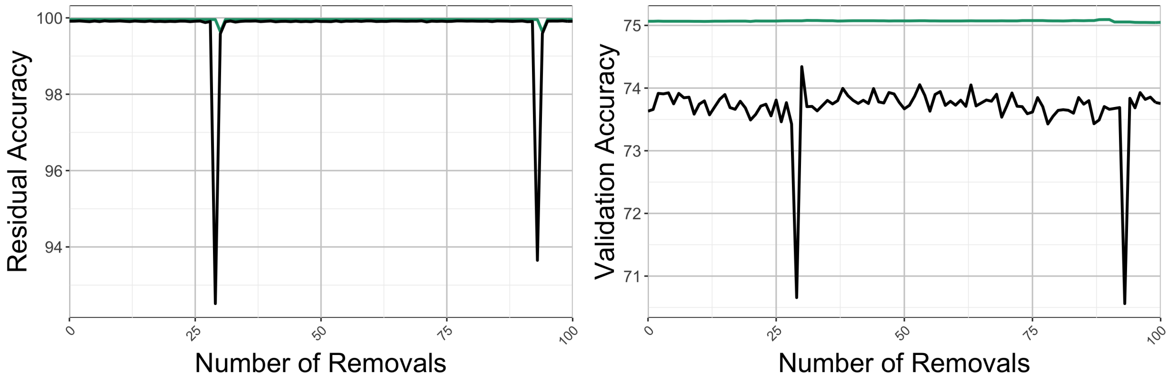


Figure B.2: Retraining Results on CIFAR. Dips occur at removal counts where the modulus equals 1.

including 1 and any size for which the modulus over the batch size equals 1. This causes particular problems when models include batch normalization layers: general practice in training deep neural networks includes the choice of dropping the last batch, so as to avoid issues of unbalanced batch sizes. For our setting we *cannot* drop these batches, because we explicitly want to measure and compute on networks trained with and without specific samples. While we can “skip" removals during our experimentation, this can still lead to odd behavior, see Figure B.2. The spikes are exactly congruent with points in the removal process corresponding to a final batch size of 1 for retraining. In general, care must be taken when attempting to unlearn from batchnorm models, and further work may be necessary to adequately address it, both in theory and practice.

CIFAR-10 Model Comparisons

Models were trained using Torch Hub, with a batch size of 64, learning rate of 0.1 for all models except VGG-11/bn, for which 0.01 was used. Data augmentation was NOT used, and weight decay was set to 0.01. 1000 perturbations were used for distribution approximation. Privacy parameters were set to $\epsilon = 0.1, \delta = 0.01$. Figures and numbers in the main thesis were averaged over 2 replications, for a random choice of 1000 samples to scrub.

LEDGAR DistillBERT Details

For the NLP experiments, we used a pretrained model from HuggingFace as a starting point. Specifically, we used the transformer model “distilbert-base-uncased", https://huggingface.co/distilbert-base-uncased which is a distilled version of the BERT base model, smaller and faster than BERT. It was pretrained on the same corpus in a self-supervised fashion, using the BERT base model as a teacher. DistilBERT Sanh et al. (2019) was pretrained on the raw texts only, without any human labels. The three losses used for the pre-training are that of distillation loss, masked language modelling and cosine embedding loss. This pre-trained model was then fine-tuned for the downstream task of provision classification using the LEDGAR dataset introduced in (Tuggener et al., 2020). We used the prototypical dataset which had 13 most common labels based on frequency. The model was fine-tuned for 4 epochs, updating all of it’s parameters without any freezing based on binary cross entropy loss with class weighting. The labels were converted to one-hot vectors and hence binary cross entropy loss was used. Learning rate used was $5e^{-5}$ and weight decay of 0.01. No weight decay was applied for bias and normalization layer parameters. We used batch size of 256 and restricted the maximum length of tokens to 128 per data point. Further gradients were clipped based on the infinity norm to a value of 1.0. We used AdamW optimizer with an epsilon value of $1e^{-8}$; and the learning rate scheduler used was WarmupLinearSchedule both from PyTorch_Transformers.

For unlearning experiments on this model, we remove provisions pertaining to a specific class. We removed samples from two classes namely “Governing Laws" and “Terminations" which had the highest and lowest support respectively. We were able to removed a varying number of samples from these classes based on the selection of the privacy parameter of ϵ for scrubbing. The results are tabulated in the main part of the thesis.

VGG-Face Identification Scrubbing

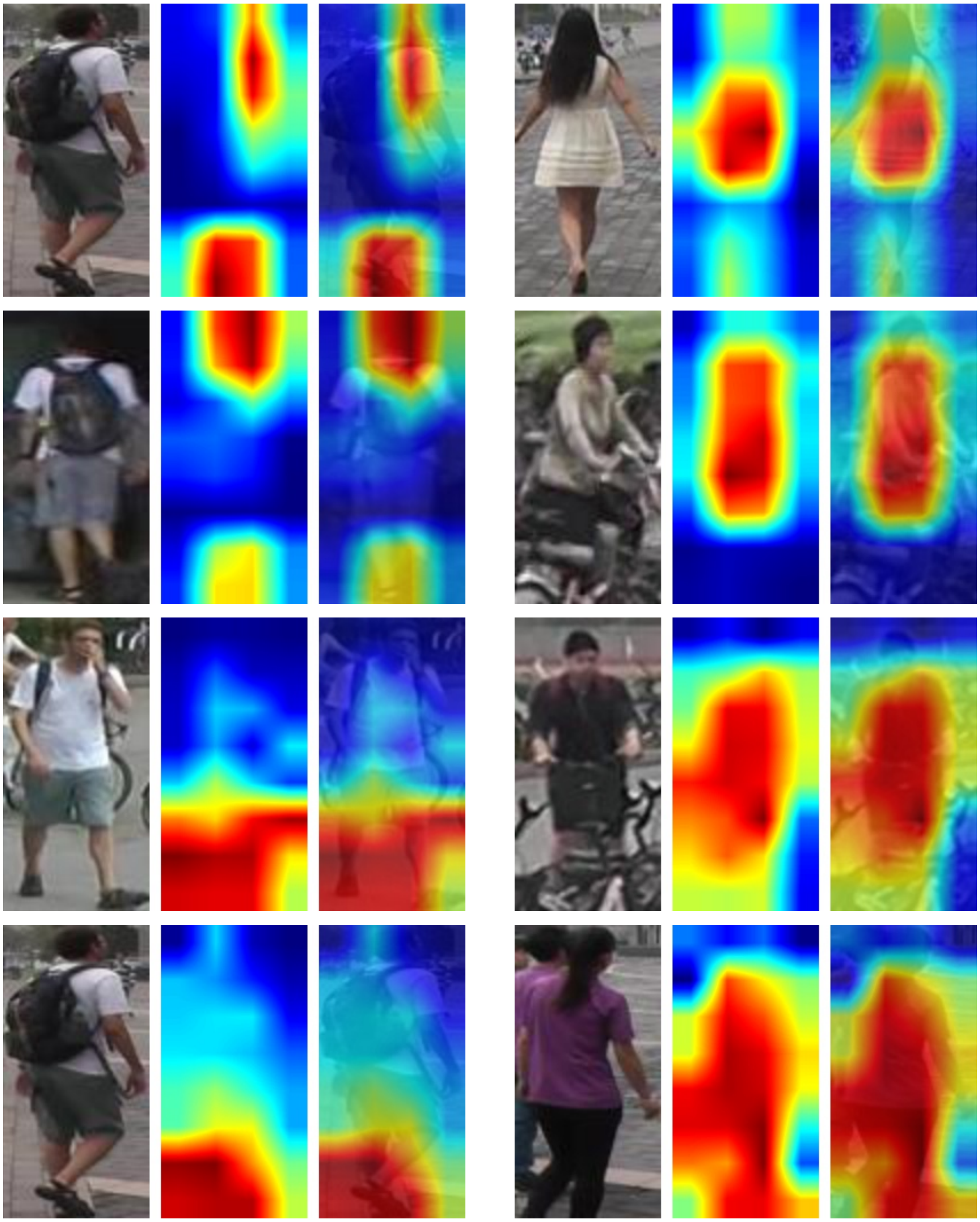
For this setting, the trained model was downloaded from https://www.robots.ox.ac.uk/~vgg/data/vgg_face/ and converted to PyTorch via https://github.com/prlz77/vgg-face.pytorch. A partial version of the dataset was constructed using the list of image URLs, consisting of 100 images for each identity within the set. The images were processed as described in the original paper (Parkhi et al., 2015).

Fine tuning was done for 4 epochs to estimate the Hessian for the sample downloaded using SGD with a learning rate of 0.0001 and a weight decay/l_2 regularization of 0.01, with a batch size of 16.

For unlearning, 100 images for a specific identity were randomly ordered and removed with $\epsilon = 0.0001, \delta = 0.01$. 100 perturbations were used to estimate the activation and loss distributions for L-FOCI.

Person Re-identification

We discuss in more detail the experimental details of unlearning deep neural networks for the person re-identification task in this section. We consider four different datasets namely, Market1501 (Zheng et al., 2015), MSMT17 (Wei et al., 2018), PRID (Hirzer et al., 2011) and QMUL GRID (Loy et al., 2009). We unlearn from different deep neural networks including ResNet50, a variant of ResNet50 with a fully connected layer (called Resnet50_fc512), Multi-Level Factorisation Net (MLFN) and MobilNet_V2. In all cases the models were first trained to reasonable accuracy as per benchmarks before proceeding with unlearning a randomly selected individual's identity from the corresponding dataset. To perform experiments pertaining to person re-identification we make use of the popular framework torchreid (Zhou et al., 2019). We had to make changes to the original code in order to make our procedure function correctly in this framework. We use Adam as the optimizer, a step scheduler and learning rate of 0.0003 across all person re-identification datasets and models used. We use softmax loss and weights were initialized using a model pre-trained on ImageNet in all cases. Images were resized to 256×128 before being used as input to any of the models. The number of training epochs was chosen accordingly to allow the training to have converged. Results from multiple runs involving different models, datasets and the privacy parameter (ϵ) are conclusive. With lower value of ϵ, e.g. 0.0005, the number of samples that could be unlearned for a particular class while maintaining model performance was lower than what could be unlearned for a higher value of the privacy parameter ϵ, e.g. 0.1. For the smaller datasets, i.e. PRID and QMUL GRID, which have approximately 2 samples per class, the unlearning procedure lead to more drastic changes as expected and it could be observed that our selection procedure selected many more layers to update than what it did for the larger datasets. Activation maps from some experiments are presented in Figure B.3.



(a) Scrubbed class

(b) Unscrubbed class

Figure B.3: Activation maps from different models (top two rows MLFN, bottom two MobileNet_V2; both trained on Market_1501) scrubbed for the person on the left. For each triplet, from (L to R) are the original image, the activation map and its image overlay. Activations change significantly for the scrubbed sample (compare column 2 to 3) whereas remain stable for the non-scrubbed sample (compare column 5 to 6).

B.3 Conditional Independence and Parameter Selection via L-CODEC

Our algorithm is directly adapted from Azadkia and Chatterjee (2019) to our parameter selection setting. Algorithm 6 shows the procedure, described for arbitrary random variables in Section 5 of Azadkia and Chatterjee (2019).

While tests for independence exist, CODEC directly estimates explanation of variance with and without the conditional variable(s) of interest. For readers interested in conditional independence more generally, and statistical and theoretical foundations, please see Spirtes et al. (2000); Dawid (1979). More recent information-based formulations can be found in Yang et al. (2020) and references therein.

Algorithm 6: Parameter MB Identification via L-CODEC (L-FOCI)

```
**Data:** $z'$, the full parameter set $w$ indexed by $\Theta := \{1, \dots, d\}$
**Result:** Sufficient set $P \subseteq \Theta$
Identify $p \in \Theta$ that maximizes $T(z', w_p)$
Set $P = \{p\}$
**while** $T(z', w_{\Theta \setminus P} | w_P) > 0$ **do**
    Identify $p \in \Theta \setminus P$ that maximizes $T(z', w_{\Theta \setminus P} | w_{\Theta \cup s})$
    **if** $T(z', w_{\setminus P}, w_{P \cup p}) < 0$ **then**
        break
    **else**
        Append $P = P \cup p$
    **end**
**end**
```

B.4 Alternate Hessian Approximations

Typical approximations are non often non-sparse; a key focus of our proposal is a reasonably informed sparse estimation in deep unlearning: we cannot allocate both full networks and the space for an inverse for 50K+ parameters (needs 10+GB alone). For Deep unlearning specifically, our sub selection makes this possible. Diagonal modification still needs full parameter updates. However, we explored the utilization of other Hessian inverse approximation schemes. More specifically, we implemented an unlearning scheme based on Kronecker-Factored Approximate Curvature (K-FAC) Martens and Grosse (2015) which exploits an efficient invertible approximation of a deep learning model's Fisher information matrix which can be non-sparse and neither low rank nor diagonal. In an experimental setup, we perform unlearning based on K-FAC from an multi layer perceptron model trained on MNIST dataset. We don't see any observable updates happening to the model based on validation metrics. Whereas, the exact same model with the exact same set of parameters can unlearn the same set of data-points using our proposed deep unlearning method based on LCODEC. We would like to point out that in order to unlearn from deep models using existing approximations schemes like K-FAC, we might have to re-imagine the update step. This demands further investigation. In other words our procedure may not be more broadly applicable to non-sparse general Hessian inverse approximations without an obvious CI structure.

We have included the code to compare K-FAC based unlearning with LCODEC based unlearning in the file https://github.com/vsingh-group/LCODEC-deep-unlearning/blob/main/scrub/kfac_scrub.py

In our implementation we heavily rely on the KFAC approximations of the Hessian as provided in https://github.com/cybertronai/autograd-lib. More instructions can be found in the README.

Appendix C

d-EMD Theoretical and Experimental Details

C.1 Proof of Theorem 6.4

Here, we present the proof of Theorem 4.2. The main observation in the proof invokes perturbation analysis Mangasarian and Meyer (1979); Ferris and Mangasarian (1991) of linear programs to assert that, under mild uniqueness conditions, small changes to a linear program's data does not change the linear program's optimal solution. The informal reason why such a result is possible can be explained using a short, geometric argument as follows.

The feasible set of a nontrivial linear program is a polytope, and, as a rule, an optimal solution to a linear program lives at the point where a hyperplane defined by the objective functional intersects a vertex of the polytope. A small perturbation of the hyperplane does not alter the intersecting vertex. The derivative of a linear program's optimal objective value, treated as a function of the data fed to the LP, has been described several times in prior work, and under a variety of conditions De Wolf and Smeers (2019); Freund (1985); Agueh and Carlier (2011); Mills (1957).

The second claim of Theorem 4.2 is useful because dual solutions to the generalized EMD linear program are not unique. The claim explains how one can modify a given solution (found by a direct gradient computation or else from an interior point solver, for example) so that it agrees with the solution yielded by the primal/dual greedy algorithm.

Proof of Theorem 4.2. Let z_j^*, for $j \in [d]$, denote an optimal solution to the dual linear program of equation (6.5) in the main thesis. Standard sensitivity analysis of linear programs implies that $z_j^* \in \mathbb{R}^n$ ($j \in [d]$) is also optimal for the perturbed linear program,

$$\begin{aligned} \underset{z_j \in \mathbb{R}^n, j \in [d]}{\text{maximize}} \quad & \sum_j (x_j + \varepsilon h_j)' z_j \\ \text{subject to} \quad & z_1(i_1) + \cdots + z_d(i_d) \leq c(i_1, \ldots, i_d), \end{aligned}$$

where the indices in the constraints include all $i_j \in [n]$, $j \in [d]$, $\varepsilon > 0$ is sufficiently small and $h_j \in \mathbb{R}^n$ are held fixed.

If $\phi(x_1 + \varepsilon h_1, \ldots, x_d + \varepsilon h_d)$ represents the optimal objective value of this program, then by linearity,

$$\phi(x_1 + \varepsilon h_1, \ldots, x_d + \varepsilon h_d) - \phi(x_1, \ldots, x_d) = \sum_j h_j' z_j^*.$$

Thus, we can form the directional derivative of ϕ as

$$\lim_{\varepsilon \to 0} \frac{\phi(x_1 + \varepsilon h_1, \ldots, x_d + \varepsilon h_d) - \phi(x_1, \ldots, x_d)}{\varepsilon \|h\|} = \frac{\sum_j h_j' z_j^*}{\|h\|},$$

where $\|h\|^2 := \sum_j \|h_j\|^2$. From this, it follows that

$$\nabla \phi(x_1, \ldots, x_d) = (z_1^*, z_2^*, \ldots, z_d^*).$$

This shows the first claim of the theorem.

To see the second claim, we have from Theorem 3.1 item 2 of Kline (2019) that

$$\sum_j z_j^*(i) = 0 \tag{C.1}$$

for all $i \in [n]$. Consequently, if one defines

$$\eta = (z_1^*(n)e, z_2^*(n)e, \cdots, z_d^*(n)e),$$

then since we assume that $e'x_j = 1$ for all $j \in [d]$,

$$\begin{aligned}\sum_j x'_j(z^*_j + t\eta) &= \sum_j x'_j z^*_j + t\, x'_j \eta \\ &= \sum_j x'_j z^*_j + t \sum_j z^*_j(n)\, x'_j e \\ &= \sum_j x'_j z^*_j,\end{aligned}$$

where the last equality holds by equation (C.1). This shows the second claim. □

C.2 Differentiable Histogramming

Algorithm 7: Differentiable Histograms

```
Function Initialization($n$):
    $r := 1/n$                                    // bin size
    $locs := arange(0, 1, r)$                     // bin boundaries
    return
Function Forward($acts$):
    $cdfs = \sigma(acts)$                         // compute CDFs
    $counts = []$
    for $loc$ in $locs$ do
        $dist = |cdfs - loc|$                     // dist. to boundary
        $ct = \sum_{i \in [nbins]} \mathrm{ReLU}(r - dist[i])$   // soft bucket count
        $counts.append(ct)$
    end
    $out = stack(counts)$
    $out = out/sum(out)$
    return $out$
```

While gradients are now readily available, typical ML pipelines do not have distributions or histograms predefined at outputs which can be fed directly into our EMD loss. Applying existing binning procedures over the batch to estimate histograms will break the ability to autodifferentiate: soft thresholds are necessary at bin boundaries such that samples within a bin may move smoothly as needed. Algorithm 7 provides a differentiable histogram implementation. Using a rectified linear relaxation allows for samples to have a continuous gradient towards neighboring bins.

C.3 Experimental Details

Results reported in tables in the main part of the thesis are of the form $M_{(SD)}$, where M is the mean and SD is the standard deviation calculated over replications.

Setup details. Experiments were conducted using NumPy and PyTorch on a Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz with an Nvidia Titan Xp GPU. Particular parameter settings and experimental runs can be found below.

Data and Licenses

The datasets Adult, Communities and Crime, and German datasets are all available under Creative Commons Attribution 4.0 International (CC BY 4.0) licenses via the UCI Machine Learning Dataset Repository https://archive.ics.uci.edu/ml/index.php. The CelebA dataset http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html is available for non-commercial purposes. See the website for more details.

ACS Data. The American Census Survey (ACS) has recently made available a large set of demographic data. The original UCI Adult dataset (Dua and Graff, 2017) was curated from this data, however recent work by Ding et al. (2021) has identified temporal shifts in demographic data, and recommends using a more recent collection as a baseline when evaluating biases and adjusting for fairness. Part of their contribution includes APIs to directly interface with the data provided by the ACS, and the ability to identify and construct similar problems associated with the original UCI-provided dataset, albeit with updated data. Data for the income prediction task was downloaded from 2018, localized to Louisiana. Race is the provided group label, which we wish to be agnostic towards, over some measure of our output. Data was accessed using the folktables codebase https://github.com/zykls/folktables with MIT License. The US Census data accessed is available for use so long as it is not used in combination with other data “to identify any particular respondent to a Census Bureau survey." See https://www.census.gov/data/developers/about/terms-of-service.html for more details.

German Data. The German dataset classifies people as good or bad credit risks. There are about 20 features (7 numerical and 13 categorical). These features represent the economic status of the person, such as, credit history, savings account, year of present employment, property and others.

Adult Data. The Adult dataset is comprised of demographic characteristics from the UCI repository (Dua and Graff, 2017). The protected attribute here is gender. It contains $44,842$ samples. The features that were used in the experiments include “age", “workclass", “fnlwgt", “education", “education-num", “marital-status", “occupation", “relationship", “race", “sex", “capital-gain", “capital-loss", “hours-per-week", “native-country", “income". A positve target label in this dataset is indicated by the attribute “income-bracket" being above $50K.

Communities and Crime Data. The Communities and Crime dataset consists of summary statistics of per-capita measures from a wide variety of communities across the United States, measured from a number of US census and surveys from the 1990's. The original goal of the dataset was to predict crime rates in communities as a function of various demographic, socioeconomic, and other features. The dataset has widely become known as the prototypical example in which using racial population distributions can be extremely harmful in perpetuating stereotypes and lead to models that continue to exacerbate inequity that may exist within the data. As such, the dataset has become a de facto tool in evaluating fairness metrics and methods that attempt to account for these inequities and biases. The preprocessed data contains 1994 samples, and we attempt to predict the violent crime rate (binarized at 0.3 after normalization between 0 and 1), and the sensitive attribute is a similarly binarized version of the percentage black population variable.

CelebA Data. CelebA (Liu et al., 2015) consists of 200K celebrity face images from the internet annotated by a group of paid adult participants. There are up to 40 labels available in the dataset, each of which is binary-valued.

Fairness Experiment Details

All experiments related to fairness use a three-layer fully-connected neural network classifier with a hidden layer size of 100. Numbers reported in Table 6.1 are means and standard deviations over three replicate runs with different seeds. Hyperparameters were selected as described in the main thesis, by taking the best result for each dataset

Table C.1: **Fairness Experiments.** Measures evaluated using standard metrics: maximum Demographic Parity Gap (**|DP|**), maximum Equalized Odds Gap (**|EO|**), and (**DEMD**). For all measures, lower values are preferred. With comparable accuracy, DEMD regularization leads to fairer representations as measured by common metrics. DP and EO measures are scaled by 100 for ease of presentation. Best results shown in bold.

		German			Adult		
Reg. Type	λ	|DP|	|EO|	DEMD	|DP|	|EO|	DEMD
None	0	$0.17_{(0.05)}$	$0.11_{(0.02)}$	$1.69_{(0.32)}$	$0.18_{(0.01)}$	$0.13_{(0.01)}$	$1.69_{(0.07)}$
DP-Reg.	0.001	$0.19_{(0.03)}$	$0.12_{(0.0)}$	$1.79_{(0.27)}$	$0.18_{(0.01)}$	$0.13_{(0.0)}$	$1.64_{(0.07)}$
	0.01	$0.27_{(0.05)}$	$0.17_{(0.01)}$	$1.5_{(0.21)}$	$0.13_{(0.01)}$	$0.14_{(0.01)}$	$0.99_{(0.09)}$
	0.1	$0.23_{(0.2)}$	$0.12_{(0.1)}$	$0.99_{(0.86)}$	$0.02_{(0.03)}$	$0.15_{(0.02)}$	$0.28_{(0.09)}$
	1.0	$0.0_{(0.0)}$	$0.19_{(0.17)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.2_{(0.0)}$	$0.0_{(0.0)}$
	10.0	$0.0_{(0.0)}$	$0.1_{(0.17)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.2_{(0.0)}$	$0.0_{(0.0)}$
	100.0	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.2_{(0.0)}$	$0.0_{(0.0)}$
EO-Reg.	0.001	$0.17_{(0.05)}$	$0.1_{(0.02)}$	$1.51_{(0.34)}$	$0.18_{(0.01)}$	$0.13_{(0.01)}$	$1.67_{(0.07)}$
	0.01	$0.14_{(0.07)}$	$0.09_{(0.04)}$	$1.41_{(0.35)}$	$0.15_{(0.01)}$	$0.12_{(0.01)}$	$1.44_{(0.08)}$
	0.1	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.47_{(0.21)}$	$0.03_{(0.01)}$	$0.06_{(0.01)}$	$0.29_{(0.02)}$
	1.0	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.08_{(0.14)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$
	10.0	$0.0_{(0.0)}$	$0.19_{(0.17)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$
	100.0	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.2_{(0.0)}$	$0.0_{(0.0)}$
Bary	0.001	$0.17_{(0.05)}$	$0.11_{(0.02)}$	$1.69_{(0.32)}$	$0.18_{(0.01)}$	$0.13_{(0.01)}$	$1.69_{(0.07)}$
	0.01	$0.17_{(0.05)}$	$0.11_{(0.02)}$	$1.68_{(0.31)}$	$0.18_{(0.01)}$	$0.13_{(0.01)}$	$1.69_{(0.07)}$
	0.1	$0.17_{(0.05)}$	$0.1_{(0.02)}$	$1.51_{(0.35)}$	$0.18_{(0.01)}$	$0.13_{(0.01)}$	$1.68_{(0.07)}$
	1.0	$0.16_{(0.06)}$	$0.1_{(0.03)}$	$1.5_{(0.26)}$	$0.17_{(0.01)}$	$0.13_{(0.01)}$	$1.6_{(0.07)}$
	10.0	$0.06_{(0.07)}$	$0.04_{(0.05)}$	$0.58_{(0.37)}$	$0.09_{(0.01)}$	$0.09_{(0.01)}$	$1.02_{(0.08)}$
	100.0	$0.06_{(0.11)}$	$0.03_{(0.06)}$	$0.06_{(0.11)}$	$0.0_{(0.0)}$	$0.04_{(0.0)}$	$0.39_{(0.05)}$
DEMD	0.001	$0.17_{(0.05)}$	$0.11_{(0.02)}$	$1.69_{(0.32)}$	$0.18_{(0.01)}$	$0.13_{(0.01)}$	$1.69_{(0.07)}$
	0.01	$0.17_{(0.05)}$	$0.11_{(0.02)}$	$1.69_{(0.32)}$	$0.18_{(0.01)}$	$0.13_{(0.01)}$	$1.69_{(0.07)}$
	0.1	$0.19_{(0.05)}$	$0.11_{(0.02)}$	$1.32_{(0.18)}$	$0.18_{(0.01)}$	$0.13_{(0.01)}$	$1.66_{(0.07)}$
	1.0	$0.2_{(0.09)}$	$0.11_{(0.05)}$	$1.59_{(0.46)}$	$0.14_{(0.01)}$	$0.12_{(0.01)}$	$1.43_{(0.07)}$
	10.0	$0.16_{(0.13)}$	$0.09_{(0.07)}$	$0.5_{(0.12)}$	$0.07_{(0.02)}$	$0.08_{(0.01)}$	$0.94_{(0.17)}$
	100.0	$0.06_{(0.08)}$	$0.18_{(0.16)}$	$0.06_{(0.08)}$	$0.01_{(0.01)}$	$0.01_{(0.01)}$	$0.41_{(0.05)}$

over the parameter range $\lambda \in [1.0, 0.1, 10, 0.01, 100, 0.001]$. The full results of this sweep are presented in Table C.1 and Table C.2.

The $|DP|$ and $|EO|$ measures are the gap between the corresponding conditional probability distributions when there is a single binary group attribute, and the largest gap $(max - min)$ when there are more than 2 groups. The DEMD loss is computed as described in the thesis with a discretization/bin level of 10.

Harmonization Experiment Details

For all our experiments related to harmonization, we use an encoder-decoder framework comprising of fully-connected layers. The hidden layers comprised of 64 nodes

Table C.2: **Fairness Experiments.** Measures evaluated using standard metrics: maximum Demographic Parity Gap (**|DP|**), maximum Equalized Odds Gap (**|EO|**), and (**DEMD**). For all measures, lower values are preferred. With comparable accuracy, DEMD regularization leads to fairer representations as measured by common metrics. DP and EO measures are scaled by 100 for ease of presentation. Best results shown in bold.

		Crime			ACS-Income		
Reg. Type	λ	|DP|	|EO|	DEMD	|DP|	|EO|	DEMD
None	0	$0.38_{(0.06)}$	$0.45_{(0.03)}$	$2.86_{(0.38)}$	$0.37_{(0.01)}$	$0.25_{(0.0)}$	$4.78_{(0.32)}$
DP-Reg.	0.001	$0.36_{(0.05)}$	$0.44_{(0.04)}$	$2.81_{(0.3)}$	$0.57_{(0.37)}$	$0.5_{(0.44)}$	$4.38_{(0.16)}$
	0.01	$0.35_{(0.02)}$	$0.44_{(0.03)}$	$2.61_{(0.17)}$	$0.81_{(0.33)}$	$0.76_{(0.42)}$	$3.45_{(0.71)}$
	0.1	$0.22_{(0.07)}$	$0.29_{(0.12)}$	$1.58_{(0.28)}$	$0.62_{(0.33)}$	$0.51_{(0.42)}$	$2.67_{(0.63)}$
	1.0	$0.0_{(0.0)}$	$0.41_{(0.03)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.67_{(0.58)}$	$0.0_{(0.0)}$
	10.0	$0.0_{(0.0)}$	$0.26_{(0.23)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.33_{(0.58)}$	$0.0_{(0.0)}$
	100.0	$0.0_{(0.0)}$	$0.26_{(0.23)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.33_{(0.58)}$	$0.0_{(0.0)}$
EO-Reg.	0.001	$0.38_{(0.05)}$	$0.45_{(0.03)}$	$2.85_{(0.38)}$	$0.36_{(0.01)}$	$0.25_{(0.0)}$	$4.77_{(0.32)}$
	0.01	$0.36_{(0.06)}$	$0.44_{(0.03)}$	$2.69_{(0.44)}$	$0.33_{(0.0)}$	$0.24_{(0.0)}$	$3.6_{(0.29)}$
	0.1	$0.11_{(0.07)}$	$0.39_{(0.03)}$	$0.61_{(0.23)}$	$0.03_{(0.0)}$	$0.03_{(0.0)}$	$0.53_{(0.01)}$
	1.0	$0.0_{(0.0)}$	$0.41_{(0.03)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$
	10.0	$0.0_{(0.0)}$	$0.41_{(0.03)}$	$0.0_{(0.0)}$	$0.0_{(0.0)}$	$0.33_{(0.58)}$	$0.0_{(0.0)}$
	100.0	$0.06_{(0.1)}$	$0.09_{(0.15)}$	$0.5_{(0.87)}$	$0.0_{(0.0)}$	$0.33_{(0.58)}$	$0.0_{(0.0)}$
Bary	0.001	$0.38_{(0.05)}$	$0.45_{(0.03)}$	$2.86_{(0.38)}$	$0.37_{(0.01)}$	$0.25_{(0.0)}$	$4.78_{(0.32)}$
	0.01	$0.38_{(0.06)}$	$0.45_{(0.03)}$	$2.86_{(0.39)}$	$0.37_{(0.01)}$	$0.25_{(0.0)}$	$4.8_{(0.32)}$
	0.1	$0.38_{(0.06)}$	$0.45_{(0.03)}$	$2.83_{(0.39)}$	$0.48_{(0.04)}$	$0.28_{(0.01)}$	$5.02_{(0.31)}$
	1.0	$0.37_{(0.06)}$	$0.44_{(0.03)}$	$2.65_{(0.45)}$	$1.0_{(0.0)}$	$1.0_{(0.0)}$	$5.66_{(0.3)}$
	10.0	$0.21_{(0.1)}$	$0.41_{(0.04)}$	$1.36_{(0.32)}$	$1.0_{(0.0)}$	$1.0_{(0.0)}$	$4.5_{(0.57)}$
	100.0	$0.01_{(0.01)}$	$0.41_{(0.03)}$	$0.19_{(0.1)}$	$0.0_{(0.0)}$	$0.33_{(0.58)}$	$0.0_{(0.0)}$
DEMD	0.001	$0.38_{(0.05)}$	$0.44_{(0.04)}$	$2.85_{(0.39)}$	$0.37_{(0.01)}$	$0.25_{(0.0)}$	$4.78_{(0.32)}$
	0.01	$0.38_{(0.05)}$	$0.45_{(0.03)}$	$2.85_{(0.38)}$	$0.38_{(0.01)}$	$0.26_{(0.0)}$	$4.82_{(0.32)}$
	0.1	$0.38_{(0.05)}$	$0.44_{(0.03)}$	$2.83_{(0.39)}$	$1.0_{(0.0)}$	$1.0_{(0.0)}$	$4.33_{(0.42)}$
	1.0	$0.35_{(0.05)}$	$0.43_{(0.03)}$	$2.54_{(0.39)}$	$0.93_{(0.12)}$	$1.0_{(0.0)}$	$2.67_{(0.17)}$
	10.0	$0.26_{(0.05)}$	$0.39_{(0.02)}$	$1.35_{(0.25)}$	$0.68_{(0.28)}$	$1.0_{(0.0)}$	$0.9_{(0.47)}$
	100.0	$0.0_{(0.0)}$	$0.41_{(0.03)}$	$0.26_{(0.02)}$	$0.0_{(0.0)}$	$1.0_{(0.0)}$	$0.0_{(0.0)}$

and the latent space was of dimension 30. We report the mean and standard deviation on unseen test datasets for three random runs. The hyper-parameter selection has been done on a validation split obtained from the training dataset. The model that achieves the best ADV (the adversarial evaluation measure), with the test set accuracy remains within 5% of the vanilla (titled "None" in the thesis) model, is chosen.

Recall that the harmonization experiments aimed at minimizing the distributional differences of the latent features across the groups. Below, we will provide details on the evaluation metrics, ADV and MMD measures, that aptly assess distributional differences of the latent features.

Evaluation Metric: Adversarial Measure (ADV) The ADV measure corresponds to the accuracy obtained by training a separate neural network to predict groups from the latent features. This step is conducted post harmonization. A lower value of the ADV accuracy denotes that the latent features are free of any group related information. This implies successful harmonization and suggests minimal distributional differences of the latent features across groups. We follow (Xie et al., 2017) for training the adversary used for reporting ADV measure. We use a three-layered fully-connected network with batch normalization and train it with Adam optimizer for 200 epochs. The learning rate for the adversary is decreased multiplicatively by a factor of 0.65 every 10 epochs for convergence.

Evaluation Metric: Maximum Mean Discrepancy (MMD) We simply use the following MMD criterion as described in (Gretton et al., 2006) and evaluate the metric on the latent features obtained from the test set. Each group is considered as a different distribution and a lower value of this metric suggests minimal distributional differences across the groups. For latent feature vector ℓ and groups i/j, we have

$$\mathcal{MMD} = \| \mathop{\mathbb{E}}_{Z_1 \sim P(\ell))_{\text{group}_i}} \mathcal{K}(Z_1, \cdot) - \mathop{\mathbb{E}}_{Z_2 \sim P(\Phi(\ell))_{\text{group}_j}} \mathcal{K}(Z_2, \cdot) \|_{\mathcal{H}} \tag{C.2}$$

The criterion is defined using a Reproducing Kernel Hilbert Space with norm $\| \cdot \|_{\mathcal{H}}$ and kernel $\mathcal{K}$.

MWGAN Details and Additional Results

The MWGAN code `https://github.com/deepmo24/MWGAN` was used under the MIT license extended from the original StarGAN codebase `https://github.com/yunjey/stargan`.

The original paper constructs an inter-domain penalty added to the multi-domain GAN discriminator loss as follows:

$$R_{MWGAN}(f) = \lambda \cdot \left(\sum_i \mathbb{E}_{\tilde{x}^{(i)} \sim \hat{\mathbb{Q}}_i} ||\nabla f(\tilde{x}^{(i)})|| - L_f \right)_+^2 \tag{C.3}$$

The sampling for expectation in practice is done by interpolating randomly between real data and generated data for each domain. In contrast to this, we directly push the gradient norms computed over samples from each domain to be close using our

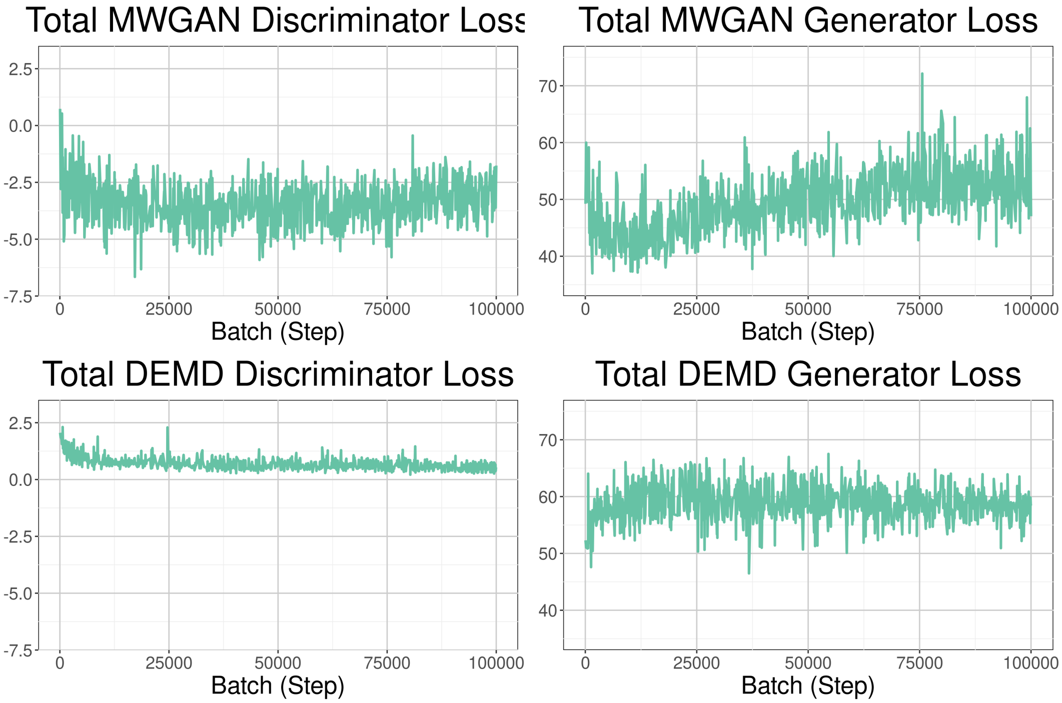


Figure C.1: Convergence Plots for MWGAN and DEMD on CelebA Multi-Domain Image Translation.

DEMD regularizer:

$$R_{DEMD}(f) = \lambda \cdot DEMD\left(\left[||\nabla f(\tilde{x}^{(i)})||\right]_i, [i]\right), \tag{C.4}$$

where $[i]$ indicates the domain associated with the fake samples generated in the same manner for each domain.

Convergence Comparison. Interestingly we observe that our construction tends to lead to more stable training. In Figure C.1, we can see that the variance of the total discriminator and generator loss fluctuate much less. We suspect that this can be directly attributed to the boundedness of the gradients of the DEMD regularizer, specific to the method where we compute the gradients via the dual of the LP: the dual variables are bounded by definition and in this particular LP, are bounded exactly by the number of bins used for discretization. While additional investigation is needed, smoothness has been identified as a desirable property of GAN training (Arora et al., 2017; Chu et al., 2020).

Model	Blond Hair	Eyeglasses	Mustache	Pale Skin
MWGAN Cao et al. (2019)	49.91	45.74	45.49	38.67
DEMD	47.29	34.43	50.69	39.60

Table C.3: Train FID Scores for CelebA Multi-Domain Image Translation.[1]

Quantitative Results. Using the Frechet Inception Distance we quantitatively measure the resulting generative samples. Following the description in Cao et al. (2019), we present the FID scores of our model and theirs in Table C.3. Our metric as a drop-in replacement performs comparably.

C.4 d-Dimensional Earth Mover's Distance Background and Algorithm

Algorithm 8 describes the greedy algorithm that solves both primal and dual generalized Earth mover's programs, also see (Kline, 2019). The algorithm accepts d distributions (i.e., histograms) $p_1, \ldots, p_d \in \mathbb{R}^n_+$ with $e'p_j = 1$ for all $j \in [d]$. Although the algorithm states that all histograms have the same number of bins, the algorithm can be easily adapted to accept as inputs $p_i \in \mathbb{R}^{n_i}_+$ with $n_i \neq n_j$. The algorithm has explicit terminal conditions for the main while loop. In the worst case the number of iterations can be nd.

Multidimensional Extensions via Slicing

Here we demonstrate the ability to drop-in replace existing schemes for computing Wasserstein-style distances when the distributions of interest exist in $\mathbb{R}^p, p > 1$. Although this manuscript focuses on analysis in the base case, empirically we find that d-EMD is able to outperform existing sinkhorn methods for multiple distributions in higher dimensions. Particularly, when the number of distributions is high and the discretization level or number of bins is reasonable, sliced sinkhorn barycenter approaches fail miserably with any practical setting of the sinkhorn regularizer weight.

[1] Results presented here computed using the pytorch-fid package with code from the MWGAN repository here: https://github.com/mseitzer/pytorch-fid. We were unable to replicate the FID scores provided in the original paper, but expect trends to be relatively similar when compared across different scoring methods.

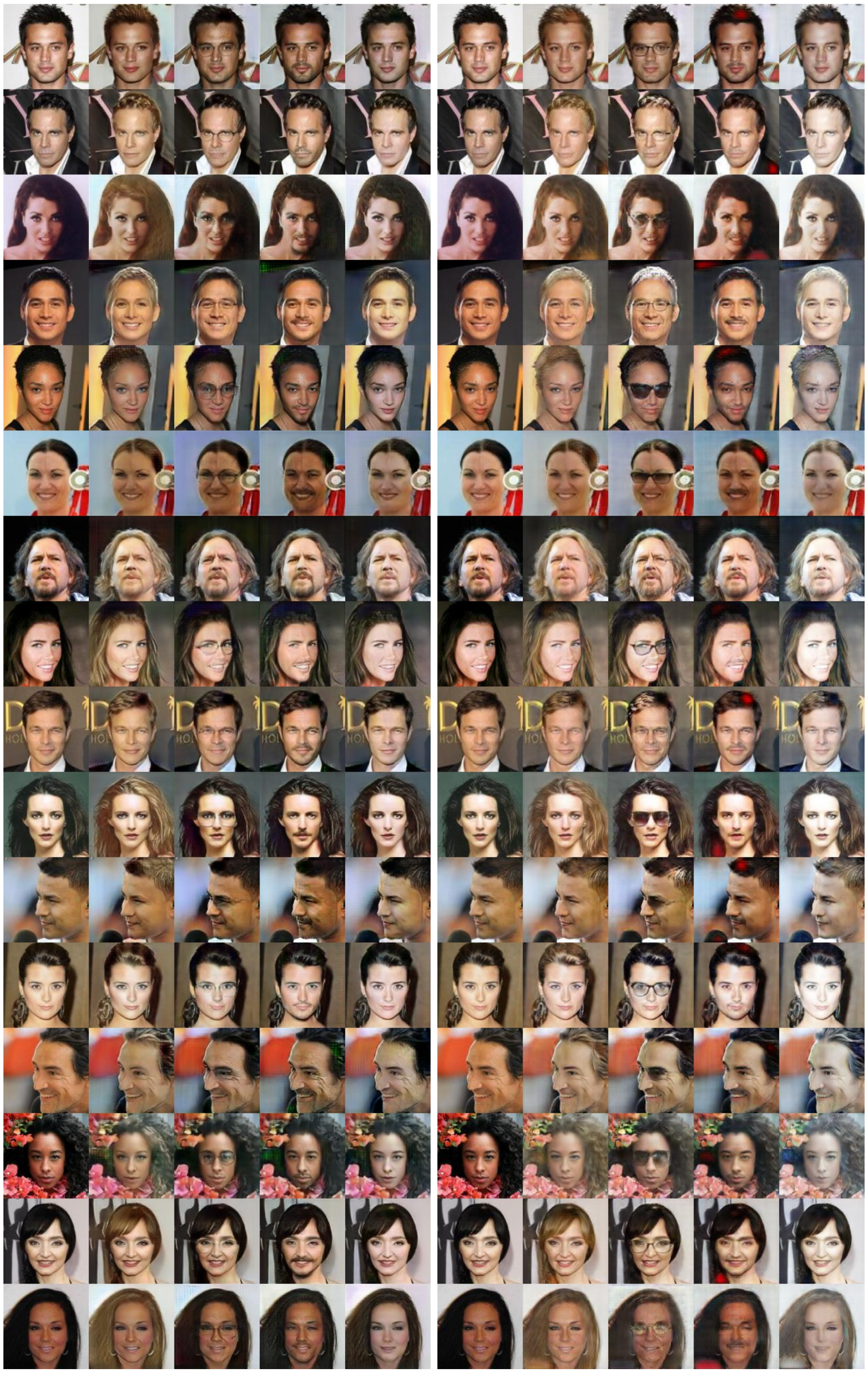


Figure C.2: More qualitative results from the multi-domain image translation problem with (Left) Cao et al. (2019), (Right) DEMD (ours). On attributes such as "Blond Hair" and "Eyeglasses", the generated images through our DEMD procedure appear more realistic. On other attributes, the generated images are comparable.

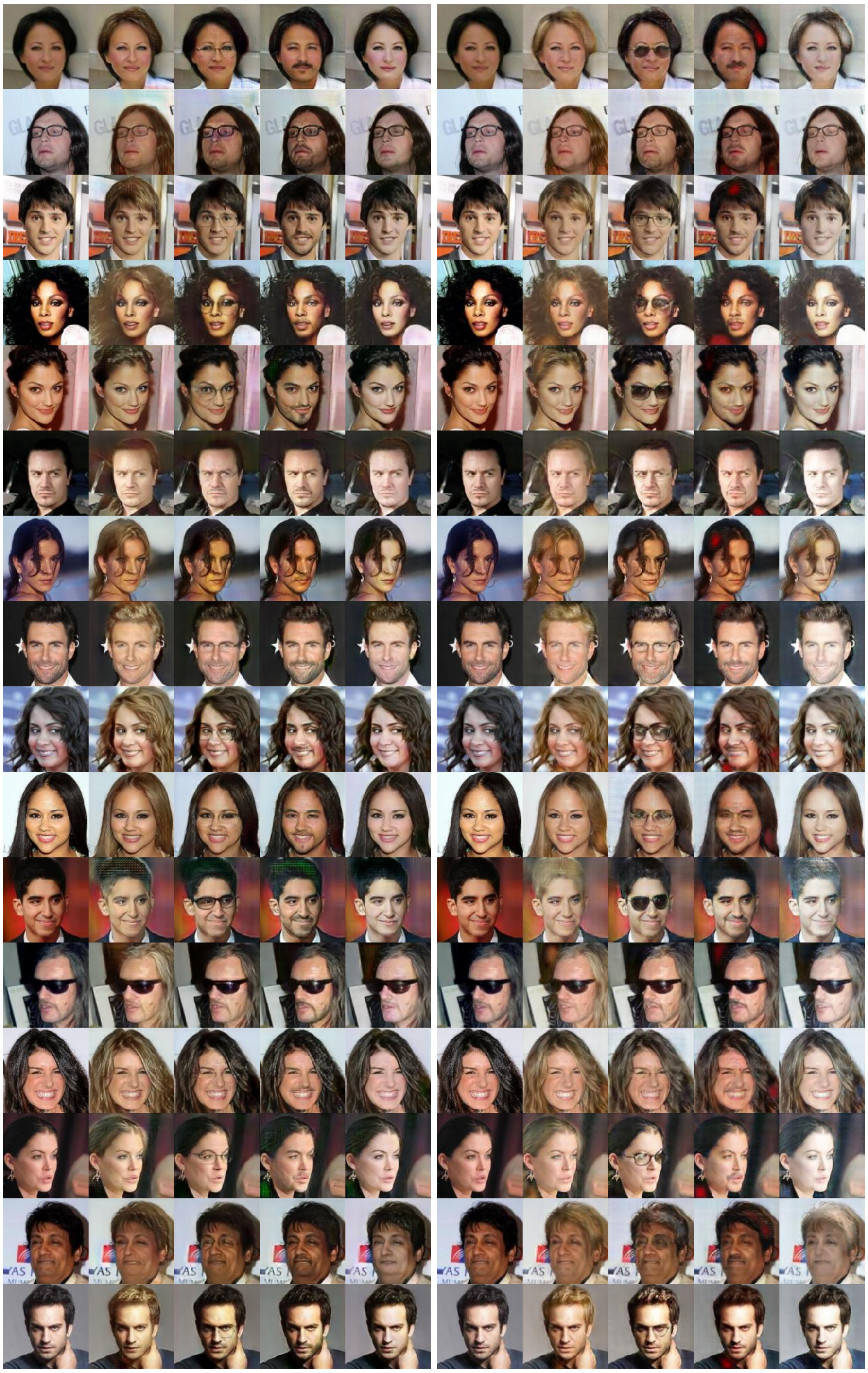


Figure C.3: More qualitative results from the multi-domain image translaton problem with (Left) Cao et al. (2019), (Right) DEMD (ours). On attributes such as “Blond Hair” and “Eyeglasses”, the generated images through our DEMD procedure appear more realistic. On other attributes the generated images are comparable.

Algorithm 8: EMD Primal/Dual Algorithm

```
Input: $p_j \in \mathbb{R}^n_+$ with $e'p_j = 1$ , $(\forall j \in [d])$
Function DEMD($n$):
    while $I(j) \leq n$, $(\forall j \in [d])$ do
        $s_k := \min_{j \in [d]} p_j(I(j))$                              // the mass to move
        $x(I) \leftarrow s_k$                                            // update the EMD solution
        $p_j(I(j)) \leftarrow p_j(I(j)) - s_k$, $(\forall j \in d)$      // shrink the data
        $j^* \leftarrow \arg\min_{j \in [d]} p_j(I(j))$
        $I(j^*) \leftarrow I(j^*) + 1$
        $k \leftarrow k + 1$
        $t_k \leftarrow c(I)$                                            // cost of this step
        if $I(j^*) \leq n$ then
            $z_{j^*}(I(j^*)) \leftarrow t_k - t_{k-1} + z_{j^*}(I(j^*) - 1)$    // update the dual solution
        end
    end
    return $x$, $(z_1, \dots, z_d)$, and the objective value $\sum_k s_k t_k$
```

The parameter-free d-EMD is able to more consistently estimate the high-dimensional distance.

C.5 An Extended Note on Ethics

As mentioned in the discussion in the main thesis body, the primary application of our proposed construction is to reduce invariance over a particular set of features. In practice, with respect to typical machine learning models and pipelines, this corresponds to minimizing performance difference as measured across subgroups within the data corresponding to a minority or protected subsets of samples or individuals. While the construction can be applied to any existing ML pipeline, we do not claim to provide a catch-all solution for group disparity that may be inherent to the data or exacerbated by the choice of the ML model that is being used. As always, care needs to be taken when working with sensitive data or models which may have disparate impacts on different groups. We point interested readers to the following extensive surveys and references therein regarding various methods and procedures for addressing and dealing with bias and unfairness in ML problems, and the potential danger associated with using models without care: (Mehrabi et al., 2021; Leavy, 2018; O'neil, 2016; d'Alessandro et al., 2017; Rakova et al., 2021).

We also note that the final multi-marginal GAN image translation application could be used to generate so-called “deepfakes." While the results of our algorithm are comparable to existing works, we believe that existing methods of identifying deepfakes would work well, and that the methods provided here and in the original paper (Cao et al., 2019) would require significant effort to be made practical for much larger scale images.

References

Agarwal, Pankaj K, Sariel Har-Peled, Kasturi R Varadarajan, et al. 2005. Geometric approximation via coresets. *Combinatorial and computational geometry* 52(1).

Agueh, Martial, and Guillaume Carlier. 2011. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis* 43(2):904–924.

Albert, Marilyn S, Steven T DeKosky, Dennis Dickson, Bruno Dubois, Howard H Feldman, Nick C Fox, Anthony Gamst, David M Holtzman, William J Jagust, Ronald C Petersen, et al. 2011. The diagnosis of mild cognitive impairment due to alzheimer's disease: Recommendations from the national institute on aging-alzheimer's association workgroups on diagnostic guidelines for alzheimer's disease. *Alzheimer's & dementia* 7(3):270–279.

Albert, Marilyn S, Mark B Moss, Rudolph Tanzi, and Kenneth Jones. 2001. Preclinical prediction of ad using neuropsychological tests. *Journal of the International Neuropsychological Society* 7(05):631–639.

Altschuler, Jason M, and Enric Boix-Adsera. 2021. Wasserstein barycenters can be computed in polynomial time in fixed dimension. *J. Mach. Learn. Res.* 22:44–1.

Alvarez-Esteban, Pedro César, Eustasio Del Barrio, Juan Antonio Cuesta-Albertos, and Carlos Matran. 2008. Trimmed comparison of distributions. *Journal of the American Statistical Association* 103(482):697–704.

Anthony, Lasse F. Wolff, Benjamin Kanding, and Raghavendra Selvan. 2020. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems. ArXiv:2007.03051.

Arias-Castro, E., E. J. Candès, et al. 2011. Detection of an anomalous cluster in a network. *The Annals of Statistics* 278–304.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th international conference on machine learning*, ed. Doina Precup and Yee Whye Teh, vol. 70 of *Proceedings of Machine Learning Research*, 214–223. PMLR.

Arora, Sanjeev, and Boaz Barak. 2009. *Computational complexity: a modern approach*. Cambridge University Press.

Arora, Sanjeev, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. 2017. Generalization and equilibrium in generative adversarial nets (GANs). In *Proceedings of the 34th international conference on machine learning*, ed. Doina Precup and Yee Whye Teh, vol. 70 of *Proceedings of Machine Learning Research*, 224–232. PMLR.

Ashburner, John, Gareth Barnes, C Chen, Jean Daunizeau, Guillaume Flandin, Karl Friston, Stefan Kiebel, James Kilner, Vladimir Litvak, Rosalyn Moran, et al. 2014. Spm12 manual. *Wellcome Trust Centre for Neuroimaging, London, UK*.

Azadkia, Mona, and Sourav Chatterjee. 2019. A simple measure of conditional dependence. *arXiv preprint arXiv:1910.12327*.

Bakator, Mihalj, and Dragica Radosav. 2018. Deep learning and medical diagnosis: A review of literature. *Multimodal Technologies and Interaction* 2(3):47.

Balikas, Georgios, Charlotte Laclau, Ievgen Redko, and Massih-Reza Amini. 2018. Cross-lingual document retrieval using regularized wasserstein distance. In *European conference on information retrieval*, 398–410. Springer.

Banerjee, M., R. Chakraborty, et al. 2015. Nonlinear regression on Riemannian manifolds and its applications to neuro-image analysis. In *Miccai*, 719–727.

Basu, Samyadeep, Phil Pope, and Soheil Feizi. 2021. Influence functions in deep learning are fragile. In *International conference on learning representations*.

Bau, David, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017a. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 6541–6549.

———. 2017b. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 6541–6549.

Bein, Wolfgang W., Peter Brucker, James K. Park, and Pramod K. Pathak. 1995. A monge property for the d-dimensional transportation problem. *Discrete Applied Mathematics* 58(2):97–109. Workshop on Discrete Algoritms.

Belilovsky, Eugene, Gaël Varoquaux, and Matthew B Blaschko. 2015. Hypothesis testing for differences in gaussian graphical models: Applications to brain connectivity. *arXiv preprint arXiv:1512.08643*.

Benamou, Jean-David, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. 2015. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing* 37(2):A1111–A1138.

Bingham, Ella, and Heikki Mannila. 2001. Random projection in dimensionality reduction: applications to image and text data. In *Sigkdd ickddm*.

Bird, Sarah, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. 2020. Fairlearn: A toolkit for assessing and improving fairness in ai. *Microsoft, Tech. Rep. MSR-TR-2020-32*.

Bonneel, Nicolas, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. 2014. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision* 51.

Bourtoule, Lucas, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 ieee symposium on security and privacy (sp)*, 141–159. IEEE.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33:1877–1901.

Cabello, Sergio, Panos Giannopoulos, Christian Knauer, and Günter Rote. 2008. Matching point sets with respect to the earth mover's distance. *Computational Geometry* 39(2):118–133.

Cai, T., W. Liu, et al. 2011. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *JASA* 106(494):594–607.

Cao, Jiezhang, Langyuan Mo, Yifan Zhang, Kui Jia, Chunhua Shen, and Mingkui Tan. 2019. Multi-marginal wasserstein gan. *Advances in Neural Information Processing Systems* 32.

Cao, Yinzhi, and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 ieee symposium on security and privacy*, 463–480. IEEE.

Carlini, Nicholas, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Shuang Song, Abhradeep Thakurta, and Florian Tramer. 2020. An attack on instahide: Is private learning possible with instance encoding? *arXiv preprint arXiv:2011.05315*.

Carlini, Nicholas, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} security symposium ({USENIX} security 19)*, 267–284.

Carlini, Nicholas, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *Usenix security symposium*, vol. 6.

Carroll, J Douglas, and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika* 35(3).

Chan, Hock Peng, and Guenther Walther. 2013. Detection with the scan and the average likelihood ratio. *Statistica Sinica* 409–428.

Chatterjee, Sourav. 2020. A new coefficient of correlation. *Journal of the American Statistical Association* 0(0):1–21. https://doi.org/10.1080/01621459.2020.1758115.

Chu, Casey, Kentaro Minami, and Kenji Fukumizu. 2020. Smoothness and stability in gans. In *International conference on learning representations*.

Chzhen, Evgenii, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. 2020. Fair regression with wasserstein barycenters. In *Advances in neural* *information processing systems*, ed. H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, vol. 33, 7321–7331. Curran Associates, Inc.

Cohen, Nadav, Or Sharir, and Amnon Shashua. 2016. On the expressive power of deep learning: A tensor analysis. In *Conference on learning theory*, 698–728.

Cohen, Nadav, and Amnon Shashua. 2016. Convolutional rectifier networks as generalized tensor decompositions. In *International conference on machine learning*, 955–963.

Collins, Maxwell D, Ji Liu, Jia Xu, Lopamudra Mukherjee, and Vikas Singh. 2014. Spectral clustering with a convex regularizer on millions of images. In *European conference on computer vision*, 282–298. Springer.

Corder, EH, AM Saunders, WJ Strittmatter, DE Schmechel, PC Gaskell, GWet al Small, AD Roses, JL Haines, and M Al Pericak-Vance. 1993. Gene dose of apolipoprotein e type 4 allele and the risk of alzheimer's disease in late onset families. *Science* 261(5123):921–923.

Cornea, E., H. Zhu, et al. 2016. Regression models on Riemannian symmetric spaces. *JRSS-B*.

Courty, Nicolas, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. 2016. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence* 39(9):1853–1865.

Creager, Elliot, David Madras, Jörn-Henrik Jacobsen, Marissa Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. 2019. Flexibly fair representation learning by disentanglement. In *International conference on machine learning*, 1436–1445. PMLR.

Cutler, Adele, and Leo Breiman. 1994. Archetypal analysis. *Technometrics* 36(4): 338–347.

Cuturi, Marco. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26:2292–2300.

Cuturi, Marco, and Arnaud Doucet. 2014. Fast computation of wasserstein barycenters. In *International conference on machine learning*, 685–693. PMLR.

Cuturi, Marco, Olivier Teboul, Jonathan Niles-Weed, and Jean-Philippe Vert. 2020. Supervised quantile normalization for low-rank matrix factorization. In *Proceedings of the 37th international conference on machine learning*, 2269–2279. Vienna, Austria.

Cybenko, George. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* 2(4):303–314.

d'Alessandro, Brian, Cathy O'Neil, and Tom LaGatta. 2017. Conscientious classification: A data scientist's guide to discrimination-aware classification. *Big data* 5(2): 120–134.

Danaher, P., P. Wang, et al. 2014. The joint graphical LASSO for inverse covariance estimation across multiple classes. *JRSS-B* 76(2):373–397.

Darst, Burcu F, Rebecca L Koscik, Annie M Racine, Jennifer M Oh, Rachel A Krause, Cynthia M Carlsson, Henrik Zetterberg, Kaj Blennow, Bradley T Christian, Barbara B Bendlin, et al. 2017. Pathway-specific polygenic risk scores as predictors of amyloid-β deposition and cognitive function in a sample at increased risk for alzheimer's disease. *Journal of Alzheimer's Disease* 55(2):473–484.

Dawid, A Philip. 1979. Conditional independence in statistical theory. *Journal of the Royal Statistical Society: Series B (Methodological)* 41(1):1–15.

De Wolf, Daniel, and Yves Smeers. 2019. Generalized derivatives of the optimal value of a linear program with respect to matrix coefficients. *European Journal of Operational Research* 291.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Diakonikolas, Jelena, and Lorenzo Orecchia. 2019. The approximate duality gap technique: A unified theory of first-order methods. *SIAM Journal on Optimization* 29(1):660–689.

Diamond, Steven, and Stephen Boyd. 2016. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research* 17(1): 2909–2913.

Ding, Frances, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems* 34.

Do Carmo, M. P. 1992. *Riemannian geometry*. Springer.

Donahue, Jeffrey, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Cvpr*, 2625–2634.

Doran, Gary, Krikamol Muandet, Kun Zhang, and Bernhard Schölkopf. 2014. A permutation-based kernel conditional independence test. In *Uai*, 132–141.

Douaud, Gwenaëlle, Saâd Jbabdi, Timothy EJ Behrens, Ricarda A Menke, Achim Gass, Andreas U Monsch, Anil Rao, Brandon Whitcher, Gordon Kindlmann, Paul M Matthews, et al. 2011. Dti measures in crossing-fibre areas: increased diffusion anisotropy reveals early white matter alteration in mci and mild alzheimer's disease. *Neuroimage* 55(3):880–890.

Du, J., A. Goh, S. Kushnarev, and A. Qiu. 2014. Geodesic regression on orientation distribution functions with its application to an aging study. *NeuroImage* 87:416–426.

Dua, Dheeru, and Casey Graff. 2017. UCI machine learning repository.

Edelman, A., T. A. Arias, and S. T. Smith. 1998. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications* 20(2): 303–353.

Elhage, Nelson, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Elsken, Thomas, Jan Hendrik Metzen, and Frank Hutter. 2019. Neural architecture search: A survey. *The Journal of Machine Learning Research* 20(1):1997–2017.

Elvander, Filip, Isabel Haasler, Andreas Jakobsson, and Johan Karlsson. 2020. Multi-marginal optimal transport using partial information with applications in robust localization and sensor fusion. *Signal Processing* 171:107474.

Erickson, William Q. 2020. A generalization for the expected value of the earth mover's distance. *arXiv preprint arXiv:2009.12723*.

Fan, J., X. Han, et al. 2012. Control of the fdr under arbitrary covariance dependence. *JASA*.

Fazlyab, Mahyar, Alexander Robey, Hamed Hassani, Manfred Morari, and George J Pappas. 2019. Efficient and accurate estimation of lipschitz constants for deep neural networks. In *Advances in neural information processing systems (neurips)*.

Ferris, Michael C, and Olvi L Mangasarian. 1991. Finite perturbation of convex programs. *Applied Mathematics and Optimization* 23(1):263–273.

Flamary, Rémi, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. 2021. Pot: Python optimal transport. *Journal of Machine Learning Research* 22(78):1–8.

Fletcher, P T. 2013. Geodesic regression and the theory of least squares on Riemannian manifolds. *IJCV* 105(2):171–185.

Fletcher, T.P., and S. Joshi. 2007. Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Processing* 87(2):250–262.

Fong, Ruth, and Andrea Vedaldi. 2018. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 8730–8738.

Fonov, VS, AC Evans, RC McKinstry, CR Almli, and DL Collins. 2009. Unbiased non-linear average age-appropriate brain templates from birth to adulthood. *NeuroImage* 47:S102. Organization for Human Brain Mapping 2009 Annual Meeting.

Frankle, Jonathan, and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International conference on learning representations*.

Fratiglioni, Laura, M Grut, Y Forsell, M Viitanen, M Grafström, K Holmen, K Ericsson, L Bäckman, A Ahlbom, and B Winblad. 1991. Prevalence of alzheimer's disease and other dementias in an elderly urban population relationship with age, sex, and education. *Neurology* 41(12):1886–1886.

Freund, Robert M. 1985. Postoptimal analysis of a linear program under simultaneous changes in matrix coefficients. In *Mathematical programming essays in honor of george b. dantzig part i*, 1–13. Springer.

Friedman, J., T. Hastie, et al. 2008. Sparse inverse covariance estimation with the graphical LASSO. *Biostatistics* 9(3):432–441.

Frogner, Charlie, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. 2015. Learning with a wasserstein loss. In *Neural information processing systems*.

FTC. 2021. California company settles ftc allegations it deceived consumers about use of facial recognition in photo storage app.

Gal, Yarin, and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Icml*, 1050–1059.

Geer, S.A. 2000. *Empirical processes in m-estimation*, vol. 6. Cambridge university press.

Geiger, Atticus, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. 2022. Inducing causal structure for interpretable neural networks. In *Proceedings of the 39th international conference on machine learning*, ed. Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, vol. 162 of *Proceedings of Machine Learning Research*, 7324–7338. PMLR.

Ginart, A, M Guan, G Valiant, and J Zou. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*.

Glaz, Joseph, Joseph I Naus, Sylvan Wallenstein, Sylvan Wallenstein, and Joseph I Naus. 2001. *Scan statistics*. Springer.

Golatkar, Aditya, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. 2021. Mixed-privacy forgetting in deep networks. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (cvpr)*, 792–801.

Golatkar, Aditya, Alessandro Achille, and Stefano Soatto. 2020a. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 9304–9312.

———. 2020b. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *European conference on computer vision*, 383–398. Springer.

Gorbunov, Eduard, Filip Hanzely, and Peter Richtárik. 2020. A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent. In *International conference on artificial intelligence and statistics*, 680–690. PMLR.

Gordaliza, Paula, Eustasio Del Barrio, Gamboa Fabrice, and Jean-Michel Loubes. 2019. Obtaining fairness using optimal transport theory. In *International conference on machine learning*, 2357–2365. PMLR.

Gou, Jianping, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129(6):1789–1819.

Gower, Robert Mansel, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. 2019. Sgd: General analysis and improved rates. In *International conference on machine learning*, 5200–5209. PMLR.

Grenander, Ulf. 2008. *Probabilities on algebraic structures*. Courier Corporation.

Grenander, Ulf, and Michael I Miller. 1998. Computational anatomy: An emerging discipline. *Quarterly of applied mathematics* 56(4):617–694.

Grenander, Ulf, and Murray Rosenblatt. 1957. Statistical analysis of stationary time series.

Gretton, Arthur, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. 2006. A kernel method for the two-sample-problem. *Advances in neural information processing systems* 19.

Guo, Chuan, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. 2020. Certified data removal from machine learning models. In *International conference on machine learning*, 3832–3842. PMLR.

Haker, Steven, Lei Zhu, Allen Tannenbaum, and Sigurd Angenent. 2004. Optimal mass transport for registration and warping. *International Journal of computer vision* 60(3):225–240.

Hardle, Wolfgang, and Enno Mammen. 1993. Comparing nonparametric versus parametric regression fits. *The Annals of Statistics* 1926–1947.

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

Hardy, John, and Dennis J Selkoe. 2002. The amyloid hypothesis of alzheimer's disease: progress and problems on the road to therapeutics. *Science* 297(5580): 353–356.

Hardy, John A, and Gerald A Higgins. 1992. Alzheimer's disease: the amyloid cascade hypothesis. *Science* 256(5054):184.

Harshman, Richard A. 1970. Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis.

Harvey, Jules., Adam. LaPlace. 2021. Exposing.ai.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 770–778.

Helfrich, Kyle, Devin Willmott, and Qiang Ye. 2017. Orthogonal recurrent neural networks with scaled cayley transform. *arXiv preprint arXiv:1707.09520*.

Higgins, Irina, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Hirzer, Martin, Csaba Beleznai, Peter M. Roth, and Horst Bischof. 2011. Person Re-Identification by Descriptive and Discriminative Classification. In *Proc. Scandinavian Conference on Image Analysis (SCIA)*.

Hjelm, R Devon, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *International conference on learning representations*.

Ho, Nhat, XuanLong Nguyen, Mikhail Yurochkin, Hung Hai Bui, Viet Huynh, and Dinh Phung. 2017. Multilevel clustering via wasserstein means. In *International conference on machine learning*, 1501–1509. PMLR.

Hoffman, Alan J, et al. 1963. On simple linear programming problems. In *Proceedings of symposia in pure mathematics*, vol. 7, 317–327.

Hofmann, Hans. 1994. Statlog (German Credit Data). UCI Machine Learning Repository.

Holtz, Sebastian, Thorsten Rohwedder, and Reinhold Schneider. 2012. On manifolds of tensors of fixed tt-rank. *Numerische Mathematik* 120(4).

Hong, Yi, Nikhil Singh, Roland Kwitt, and Marc Niethammer. 2015. Group testing for longitudinal data. In *International conference on information processing in medical imaging*, 139–151. Springer.

Huang, Gao, Chuan Quo, Matt J Kusner, Yu Sun, Kilian Q Weinberger, and Fei Sha. 2016. Supervised word mover's distance. In *Proceedings of the 30th international conference on neural information processing systems*, 4869–4877.

Huang, Gary B, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 2008. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'real-life'images: detection, alignment, and recognition*.

Huang, Haiwen, Zhihan Li, Lulu Wang, Sishuo Chen, Bin Dong, and Xinyu Zhou. 2020. Feature space singularity for out-of-distribution detection. *arXiv preprint arXiv:2011.14654*.

Hwang, Seong Jae, Ronak R. Mehta, Hyunwoo J. Kim, Sterling C. Johnson, and Vikas Singh. 2020a. Sampling-free uncertainty estimation in gated recurrent units with applications to normative modeling in neuroimaging. In *Proceedings of the 35th uncertainty in artificial intelligence conference*, ed. Ryan P. Adams and Vibhav Gogate, vol. 115 of *Proceedings of Machine Learning Research*, 809–819. PMLR.

———. 2020b. Sampling-free uncertainty estimation in gated recurrent units with applications to normative modeling in neuroimaging. In *Proceedings of the 35th uncertainty in artificial intelligence conference*, ed. Ryan P. Adams and Vibhav Gogate, vol. 115 of *Proceedings of Machine Learning Research*, 809–819. PMLR.

Hwang, Seong Jae, Sathya N. Ravi, Zirui Tao, Hyunwoo Kim, Maxwell D. Collins, and Vikas Singh. 2018. Tensorize, factorize and regularize: Robust visual relationship learning. In *Proceedings of ieee conference on computer vision and pattern recognition (cvpr)*.

Izzo, Zachary, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. 2021. Approximate data deletion from machine learning models. In *International conference on artificial intelligence and statistics*, 2008–2016. PMLR.

Jack Jr, Clifford R, Heather J Wiste, Prashanthi Vemuri, Stephen D Weigand, Matthew L Senjem, Guang Zeng, Matt A Bernstein, Jeffrey L Gunter, Vernon S Pankratz, Paul S Aisen, et al. 2010. Brain beta-amyloid measures and magnetic resonance imaging atrophy both predict time-to-progression from mild cognitive impairment to alzheimer's disease. *Brain* 133(11):3336–3348.

Jacot, Arthur, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems* 31.

Janati, Hicham, Marco Cuturi, and Alexandre Gramfort. 2020. Debiased sinkhorn barycenters. In *International conference on machine learning*, 4692–4701. PMLR.

Jayasumana, S., et al. 2013. Kernel methods on the Riemannian manifold of SPD matrices. In *Cvpr*.

Jeng, X Jessie, T Tony Cai, and Hongzhe Li. 2010. Optimal sparse segment identification with application in copy number variation analysis. *Journal of the American Statistical Association* 105(491):1156–1166.

Jiang, Ray, Aldo Pacchiano, Tom Stepleton, Heinrich Jiang, and Silvia Chiappa. 2020. Wasserstein fair classification. In *Uncertainty in artificial intelligence*, 862–872. PMLR.

Jordan, M.I. 1998. *Learning in graphical models*, vol. 89. Springer Science & Business Media.

Just, Hoang Anh, Feiyang Kang, Tianhao Wang, Yi Zeng, Myeongseob Ko, Ming Jin, and Ruoxi Jia. 2023. LAVA: Data valuation without pre-specified learning algorithms. In *The eleventh international conference on learning representations*.

Kaneko, T., S. Fiori, and T. Tanaka. 2013. Empirical arithmetic averaging over the compact stiefel manifold. *IEEE Transactions on Signal Processing* 61(4):883–894.

Kantorovich, Leonid V. 1942. On the translocation of masses. *Dokl. Akad. Nauk SSSR* 37:227–229.

———. 1960. Mathematical methods of organizing and planning production. *Management science* 6(4):366–422.

———. 2006. On the translocation of masses. *Journal of mathematical sciences* 133(4): 1381–1382.

Kantorovitch, Leonid. 1958. On the translocation of masses. *Management science* 5(1):1–4.

Karasuyama, Masayuki, and Ichiro Takeuchi. 2009. Multiple incremental decremental learning of support vector machines. *Advances in neural information processing systems* 22:907–915.

Karcher, Hermann. 1977. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics* 30(5):509–541.

Karras, Tero, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 4401–4410.

Kaye, Kate. 2022. The ftc's new enforcement weapon spells death for algorithms.

Kearns, Michael, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International conference on machine learning*, 2564–2572. PMLR.

Khrulkov, Valentin, Oleksii Hrinchuk, and Ivan Oseledets. 2019. Generalized tensor models for recurrent neural networks. In *International conference on learning representations*.

Kim, Hyunwoo J, Nagesh Adluru, et al. 2014. Multivariate general linear models on Riemannian manifolds with applications to statistical analysis of diffusion weighted images. In *Cvpr*, 2705–2712.

Kingma, Diederik P, and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kline, Jeffery. 2019. Properties of the d-dimensional earth mover's problem. *Discrete Applied Mathematics* 265:128–141.

Klus, Stefan, Patrick Gelß, Sebastian Peitz, and Christof Schütte. 2018. Tensor-based dynamic mode decomposition. *Nonlinearity* 31(7):3359.

Koh, Pang Wei, and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, 1885–1894. PMLR.

Koller, D., and N. Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM* 60(6):84–90.

Kwegyir-Aggrey, Kweku, Rebecca Santorella, and Sarah M. Brown. 2021. Everything is relative: Understanding fairness with optimal transport. 2102.10349.

Laurent, Beatrice, and Pascal Massart. 2000. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics* 1302–1338.

Lauritzen, S.L. 1996. *Graphical models*. Clarendon Press.

Le, Khang, Huy Nguyen, Quang M Nguyen, Tung Pham, Hung Bui, and Nhat Ho. 2021. On robust optimal transport: computational complexity and barycenter computation. *Advances in Neural Information Processing Systems* 34:21947–21959.

Leavy, Susan. 2018. Gender bias in artificial intelligence: The need for diversity and gender theory in machine learning. In *Proceedings of the 1st international workshop on gender equality in software engineering*, 14–16.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521(7553):436–444.

Lee, John M. 2003. Smooth manifolds. In *Introduction to smooth manifolds*, 1–29. Springer.

Lezak, Muriel Deutsch. 2004. *Neuropsychological assessment*. Oxford University Press, USA.

Li, Yujia, Kevin Swersky, and Richard Zemel. 2014. Learning unbiased features.

Lin, Tianyi, Nhat Ho, Marco Cuturi, and Michael I. Jordan. 2022. On the complexity of approximating multimarginal optimal transport. *Journal of Machine Learning Research* 23(65):1–43.

Lin, Yi-Cheng, Yao-Chia Shih, Wen-Yih I Tseng, Yu-Hsiu Chu, Meng-Tien Wu, Ta-Fu Chen, Pei-Fang Tang, and Ming-Jang Chiu. 2014. Cingulum correlates of cognitive functions in patients with mild cognitive impairment and early alzheimer's disease: a diffusion spectrum imaging study. *Brain topography* 27(3):393–402.

Ling, Haibin, and Kazunori Okada. 2007. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE transactions on pattern analysis and machine intelligence* 29(5):840–853.

Liu, H., J. Lafferty, et al. 2009. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *JMLR* 10:2295–2328.

Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of international conference on computer vision (iccv)*.

Locatello, Francesco, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, 4114–4124. PMLR.

Loizou, Nicolas, and Peter Richtárik. 2020. Convergence analysis of inexact randomized iterative methods. *SIAM Journal on Scientific Computing* 42(6):A3979–A4016.

Loizou, Nicolas, and Peter Richtárik. 2019. Convergence analysis of inexact randomized iterative methods. 1903.07971.

Lokhande, Vishnu Suresh, Rudrasis Chakraborty, Sathya N. Ravi, and Vikas Singh. 2022. Equivariance allows handling multiple nuisance variables when analyzing pooled neuroimaging datasets.

Loy, Chen Change, Tao Xiang, and Shaogang Gong. 2009. Multi-camera activity correlation analysis. In *2009 ieee conference on computer vision and pattern recognition*, 1988–1995. IEEE.

Lubich, Christian, Ivan V Oseledets, and Bart Vandereycken. 2015. Time integration of tensor trains. *SIAM Journal on Numerical Analysis* 53(2):917–941.

Luise, Giulia, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. 2018. Differential properties of sinkhorn approximation for learning with wasserstein distance. In *Advances in neural information processing systems*, ed. S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, vol. 31. Curran Associates, Inc.

Luise, Giulia, Saverio Salzo, Massimiliano Pontil, and Carlo Ciliberto. 2019. Sinkhorn barycenters with free support via frank-wolfe algorithm. In *Advances in neural information processing systems*, ed. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, vol. 32. Curran Associates, Inc.

Lundberg, Scott M, and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30.

Mangasarian, Olvi L, and RR Meyer. 1979. Nonlinear perturbation of linear programs. *SIAM Journal on Control and Optimization* 17(6):745–752.

Marszałek, Marcin, Ivan Laptev, and Cordelia Schmid. 2009. Actions in context. In *Ieee conference on computer vision & pattern recognition*.

Martens, James, and Roger Grosse. 2015. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, 2408–2417. PMLR.

McArdle, J.J., and R.Q. Bell. 2000. An introduction to latent growth models for developmental data analysis.

Mehrabi, Ninareh, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54(6):1–35.

Mehta, Ronak, Rudrasis Chakraborty, Yunyang Xiong, and Vikas Singh. 2019a. Scaling recurrent models via orthogonal approximations in tensor trains. In *Proceedings of the ieee/cvf international conference on computer vision (iccv)*.

Mehta, Ronak, Hyunwoo Kim, Shulei Wang, Sterling Johnson, Ming Yuan, and Vikas Singh. 2019b. Localizing differentially evolving covariance structures via scan statistics. *Quarterly of Applied Math.* 77(2):357–398.

Mehta, Ronak, Jeffery Kline, Vishnu Suresh Lokhande, Glenn Fung, and Vikas Singh. 2023. Efficient discrete multi marginal optimal transport regularization. In *The eleventh international conference on learning representations*.

Mehta, Ronak, Sourav Pal, Vikas Singh, and Sathya N. Ravi. 2022. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (cvpr)*, 10422–10431.

Mills, Harlan D. 1957. Marginal values of matrix games and linear programs. *Linear Inequalities and Related Systems* 38.

Minaee, Shervin, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. 2021. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*.

Monge, Gaspard. 1781. Memoire sur la theorie des deblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.* 666–704.

Mori, S., K. Oishi, et al. 2008. Stereotaxic white matter atlas based on diffusion tensor imaging in an icbm template. *Neuroimage* 40(2):570–582.

Mormino, Elizabeth C, Rebecca A Betensky, Trey Hedden, Aaron P Schultz, Andrew Ward, Willem Huijbers, Dorene M Rentz, Keith A Johnson, Reisa A Sperling, Alzheimer's Disease Neuroimaging Initiative, et al. 2014. Amyloid and apoe $\varepsilon 4$ interact to influence short-term decline in preclinical alzheimer disease. *Neurology* 82(20):1760–1767.

Muralidharan, Prasanna, and P Thomas Fletcher. 2012. Sasaki metrics for analysis of longitudinal data on manifolds. In *Computer vision and pattern recognition (cvpr), 2012 ieee conference on*, 1027–1034. IEEE.

Myers, Jerome L, Arnold D Well, and Robert F Lorch. 2013. *Research design and statistical analysis*. Routledge.

Naidich, Thomas P, Henri M Duvernoy, Bradley N Delman, A Gregory Sorensen, Spyros S Kollias, and E Mark Haacke. 2009. *Duvernoy's atlas of the human brain stem and cerebellum: high-field mri, surface anatomy, internal structure, vascularization and 3 d sectional anatomy*. Springer Science & Business Media.

Nazarovs, Jurijs, Ronak R. Mehta, Vishnu Suresh Lokhande, and Vikas Singh. 2021. Graph reparameterizations for enabling 1000+ monte carlo iterations in bayesian deep neural networks. In *Proceedings of the thirty-seventh conference on uncertainty in artificial intelligence*, ed. Cassio de Campos and Marloes H. Maathuis, vol. 161 of *Proceedings of Machine Learning Research*, 118–128. PMLR.

Neel, Seth, Aaron Roth, and Saeed Sharifi-Malvajerdi. 2021. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic learning theory*, 931–962. PMLR.

Neyman, Jerzy, Egon Sharpe Pearson, and Karl Pearson. 1933. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231(694-706):289–337. https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.1933.0009.

Neyshabur, Behnam, Ryota Tomioka, and Nathan Srebro. 2014. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*.

Noé, Frank, Gianni De Fabritiis, and Cecilia Clementi. 2020. Machine learning for protein folding and dynamics. *Current Opinion in Structural Biology* 60:77–84.

Novikov, Alexander, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P Vetrov. 2015. Tensorizing neural networks. In *Nips*, 442–450.

Novikov, Alexander, Mikhail Trofimov, and Ivan Oseledets. 2017. Exponential machines. *ICLR Workshop Track*.

Nowozin, Sebastian, Botond Cseke, and Ryota Tomioka. 2016. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, ed. D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, vol. 29. Curran Associates, Inc.

O'neil, Cathy. 2016. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway Books.

Orbanz, Peter. 2016. Probability theory.

Oseledets, Ivan V. 2011. Tensor-train decomposition. *SIAM Journal on Scientific Computing* 33(5):2295–2317.

Parkhi, Omkar M., Andrea Vedaldi, and Andrew Zisserman. 2015. Deep face recognition. In *British machine vision conference*.

Pass, Brendan. 2015. Multi-marginal optimal transport: theory and applications. *ESAIM: Mathematical Modelling and Numerical Analysis* 49(6):1771–1790.

Peleg, Shmuel, Michael Werman, and Hillel Rom. 1989. A unified approach to the change of resolution: Space and gray-level. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(7):739–742.

Qiu, H., F. Han, et al. 2015. Joint estimation of multiple graphical models from high dimensional time series. *JRSS-B*.

Qiu, Huitong, Fang Han, Han Liu, and Brian Caffo. 2016. Joint estimation of multiple graphical models from high dimensional time series. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78(2):487–504. https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/rssb.12123.

Rabin, Julien, Gabriel Peyré, Julie Delon, and Marc Bernot. 2011. Wasserstein barycenter and its application to texture mixing. In *International conference on scale space and variational methods in computer vision*, 435–446. Springer.

Racine, A. M, N. Adluru, et al. 2014. Associations between white matter microstructure and amyloid burden in preclinical AD: a multimodal imaging investigation. *NeuroImage: Clinical* 4:604–614.

Rakova, Bogdana, Jingying Yang, Henriette Cramer, and Rumman Chowdhury. 2021. Where responsible ai meets reality: Practitioner perspectives on enablers for shifting organizational practices. *Proceedings of the ACM on Human-Computer Interaction* 5(CSCW1):1–23.

Ravi, Sathya N, Maxwell D Collins, and Vikas Singh. 2019. A deterministic nonsmooth frank wolfe algorithm with coreset guarantees. *Informs Journal on Optimization* 1(2):120–142.

Redmond, Michael. 2009. Communities and Crime. UCI Machine Learning Repository.

Ren, Jie, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. *Advances in Neural Information Processing Systems* 32.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 1135–1144. KDD ’16, New York, NY, USA: Association for Computing Machinery.

Rimol, Lars M, Ingrid Agartz, Srdjan Djurovic, Andrew A Brown, J Cooper Roddey, Anna K Kähler, Morten Mattingsdal, Lavinia Athanasiu, Alexander H Joyner, Nicholas J Schork, et al. 2010. Sex-dependent association of common variants of microcephaly genes with brain structure. *Proceedings of the National Academy of Sciences* 107(1):384–388.

Roehrig, Charles S. 1988. Conditions for identification in nonparametric and parametric models. *Econometrica: Journal of the Econometric Society* 433–447.

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 10684–10695.

Romero, Enrique, Ignacio Barrio, and Lluís Belanche. 2007. Incremental and decremental learning for linear support vector machines. In *International conference on artificial neural networks*, 209–218. Springer.

Rubner, Yossi, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision* 40(2):99–121.

Rüschendorf, Ludger, and Ludger Uckelmann. 2002. On the n-coupling problem. *Journal of multivariate analysis* 81(2):242–258.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115(3):211–252.

Sager, Mark A, Bruce Hermann, and Asenath La Rue. 2005. Middle-aged children of persons with alzheimer's disease: Apoe genotypes and cognitive function in the wisconsin registry for alzheimer's prevention. *Journal of geriatric psychiatry and neurology* 18(4):245–249.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schmidt, Michael, et al. 1996. *Rey auditory verbal learning test: a handbook*. Western Psychological Services Los Angeles.

Schuhmann, Christoph, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

Seber, George AF, and Alan J Lee. 2003. Linear regression analysis. hoboken. *NJ: Wiley. doi* 10:9780471722199.

Sekhari, Ayush, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. 2021. Remember what you want to forget: Algorithms for machine unlearning. 2103.03279.

Selvaraju, Ramprasaath R, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the ieee international conference on computer vision*, 618–626.

Shi, Xiaoyan, Martin Styner, Jeffrey Lieberman, Joseph G Ibrahim, Weili Lin, and Hongtu Zhu. 2009. Intrinsic regression models for manifold-valued data. In *Medical image computing and computer-assisted intervention–miccai 2009: 12th international conference, london, uk, september 20-24, 2009, proceedings, part ii 12*, 192–199. Springer.

Silver, David, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature* 529(7587):484–489.

Simsekli, Umut, Lingjiong Zhu, Yee Whye Teh, and Mert Gurbuzbalaban. 2020. Fractional underdamped langevin dynamics: Retargeting sgd with momentum under heavy-tailed gradient noise. In *International conference on machine learning*, 8970–8980. PMLR.

Solomon, Justin, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. 2015. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)* 34(4):1–11.

Sommer, S., F. Lauze, et al. 2014. Optimization over geodesics for exact principal geodesic analysis. *Adv. in Comp. Math.* 40(2):283–313.

Spirtes, Peter, Clark N Glymour, Richard Scheines, and David Heckerman. 2000. *Causation, prediction, and search*. MIT press.

Spivak, M. 1981. *Comprehensive introduction to differential geometry*. Publish or Perish, Inc.

Srivastava, Nitish, Elman Mansimov, and Ruslan Salakhudinov. 2015. Unsupervised learning of video representations using lstms. In *Icml*, 843–852.

Stehr, Mark. 2007. The effect of cigarette taxes on smoking among men and women. *Health Economics* 16(12):1333–1343.

Städler, Nicolas, and Sach Mukherjee. 2012. Two-sample testing in high-dimensional models. arXiv:1210.4584.

Su, Jingyong, Sebastian Kurtek, Eric Klassen, Anuj Srivastava, et al. 2014. Statistical analysis of trajectories on riemannian manifolds: bird migration, hurricane tracking and video surveillance. *The Annals of Applied Statistics* 8(1):530–552.

Sun, Yiyou, Sathya N. Ravi, and Vikas Singh. 2019. Adaptive activation thresholding: Dynamic routing type behavior for interpretability in convolutional neural networks. In *Proceedings of the ieee/cvf international conference on computer vision (iccv)*.

Székely, Gábor J, and Maria L Rizzo. 2014. Partial distance correlation with methods for dissimilarities. *The Annals of Statistics* 42(6):2382–2412.

Talagrand, Michel. 2006. *The generic chaining: upper and lower bounds of stochastic processes*. Springer Science & Business Media.

Tanzi, Rudolph E, and Lars Bertram. 2005. Twenty years of the alzheimer's disease amyloid hypothesis: a genetic perspective. *Cell* 120(4):545–555.

Tolstikhin, Ilya, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2018. Wasserstein auto-encoders. In *International conference on learning representations*.

Tombaugh, Tom N. 2004. Trail making test a and b: normative data stratified by age and education. *Archives of clinical neuropsychology* 19(2):203–214.

Traonmilin, Yann, and Jean-François Aujol. 2020. The basins of attraction of the global minimizers of the non-convex sparse spike estimation problem. *Inverse Problems* 36(4):045003.

Tuggener, Don, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. 2020. Ledgar: a large-scale multi-label corpus for text classification of legal provisions in contracts. In *12th language resources and evaluation conference (lrec) 2020*, 1228–1234. European Language Resources Association.

Tupitsa, Nazarii, Pavel Dvurechensky, Alexander Gasnikov, and César A. Uribe. 2020. Multimarginal optimal transport by accelerated alternating minimization. In *2020 59th ieee conference on decision and control (cdc)*, 6132–6137.

Tuzel, Oncel, Fatih Porikli, and Peter Meer. 2007. Human detection via classification on Riemannian manifolds. In *Computer vision and pattern recognition, 2007. cvpr'07. ieee conference on*, 1–8. IEEE.

Ullman, J.B., and P.M. Bentler. 2003. *Structural equation modeling*. Wiley Online Library.

Vamathevan, Jessica, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. 2019. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery* 18(6):463–477.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30.

Villani, Cédric. 2009. *Optimal transport: old and new*, vol. 338. Springer.

———. 2021. *Topics in optimal transportation*, vol. 58. American Mathematical Soc.

Walther, G. 2010. Optimal and fast detection of spatial clusters with scan statistics. *The Annals of Statistics* 38(2):1010–1033.

Wang, Fan, and Leonidas J Guibas. 2012. Supervised earth mover's distance learning and its computer vision applications. In *European conference on computer vision*, 442–455. Springer.

Wang, S., J. Fan, et al. 2016. Structured correlation detection with application to colocalization analysis in dual-channel fluorescence microscopic imaging. *arXiv preprint arXiv:1604.02158*.

Wang, Xueqin, Wenliang Pan, Wenhao Hu, Yuan Tian, and Heping Zhang. 2015. Conditional distance correlation. *Journal of the American Statistical Association* 110(512): 1726–1734.

Wechsler, David. 2014. Wechsler adult intelligence scale–fourth edition (wais–iv).

Wei, Longhui, Shiliang Zhang, Wen Gao, and Qi Tian. 2018. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 79–88.

Wilks, S. S. 1938. The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses. *The Annals of Mathematical Statistics* 9(1):60 – 62.

Wragg, Robin E, and Dilip V Jeste. 1989. Overview of depression and psychosis in alzheimer's disease. *Am J Psychiatry* 146(5):577–587.

Wright, Stephen, Jorge Nocedal, et al. 1999. Numerical optimization. *Springer Science* 35(67-68):7.

Wu, G., E.Y. Chang, et al. 2005. An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines. In *Icml*, vol. 8.

Xie, Qizhe, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. 2017. Controllable invariance through adversarial feature learning. In *Advances in neural information processing systems*, ed. I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, vol. 30. Curran Associates, Inc.

Xie, Y., B.C. Vemuri, et al. 2010. Statistical analysis of tensor fields. In *Miccai*, 682–689. Springer.

Xie, Yujia, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. 2020. Differentiable top-k with optimal transport. In *Advances in neural information processing systems*, ed. H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, vol. 33, 20520–20531. Curran Associates, Inc.

Xiong, Yunyang, Hyunwoo J Kim, and Varsha Hedau. 2019a. Antnets: Mobile convolutional neural networks for resource efficient image classification. *arXiv preprint arXiv:1904.03775*.

Xiong, Yunyang, Hyunwoo J. Kim, Bhargav Tangirala, Ronak Mehta, Sterling C. Johnson, and Vikas Singh. 2019b. On training deep 3d cnn models with dependent samples in neuroimaging. In *Information processing in medical imaging*, ed. Albert C. S. Chung, James C. Gee, Paul A. Yushkevich, and Siqi Bao, 99–111. Cham: Springer International Publishing.

Xiong, Yunyang, Hanxiao Liu, Suyog Gupta, Berkin Akin, Gabriel Bender, Yongzhe Wang, Pieter-Jan Kindermans, Mingxing Tan, Vikas Singh, and Bo Chen. 2021. Mobiledets: Searching for object detection architectures for mobile accelerators. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 3825–3834.

Xiong, Yunyang, Ronak Mehta, and Vikas Singh. 2019c. Resource constrained neural network architecture search: Will a submodularity assumption help? In *Proceedings of the ieee/cvf international conference on computer vision (iccv)*.

Xue, L., and H.and others Zou. 2012. Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *The Annals of Statistics* 40(5):2541–2571.

Yang, Alan, AmirEmad Ghassami, Maxim Raginsky, Negar Kiyavash, and Elyse Rosenbaum. 2020. Model-augmented conditional mutual information estimation for feature selection. In *Conference on uncertainty in artificial intelligence*, 1139–1148. PMLR.

Yang, S., Z. Lu, et al. 2015. Fused multiple graphical lasso. *SIAM J. Opt.* 25(2): 916–943.

Yang, Yinchong, Denis Krompass, and Volker Tresp. 2017. Tensor-train recurrent neural networks for video classification. In *Icml*.

Ye, Jieping, Ravi Janardan, and Qi Li. 2005. Two-dimensional linear discriminant analysis. In *Nips*, 1569–1576.

Yeung, Daniel S, Ian Cloete, Daming Shi, and Wing wY Ng. 2010. *Sensitivity analysis for neural networks*. Springer.

Yu, Fisher, and Vladlen Koltun. 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Yu, Xiyu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. 2017. On compressing deep models by low rank and sparse decomposition. In *Cvpr*, 7370–7379.

Yuan, M. 2010. High dimensional inverse covariance matrix estimation via LP. *JMRL* 11:2261–2286.

Yuan, Ying, Hongtu Zhu, Weili Lin, and James Stephen Marron. 2012. Local polynomial regression for symmetric positive definite matrices. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74(4):697–719.

Yurochkin, Mikhail, Sebastian Claici, Edward Chien, Farzaneh Mirzazadeh, and Justin Solomon. 2019. Hierarchical optimal transport for document representation. 1906.10827.

Zeng, Zhanpeng, Yunyang Xiong, Sathya Ravi, Shailesh Acharya, Glenn M Fung, and Vikas Singh. 2021. You only sample (almost) once: Linear cost self-attention via bernoulli sampling. In *Proceedings of the 38th international conference on machine learning*, ed. Marina Meila and Tong Zhang, vol. 139 of *Proceedings of Machine Learning Research*, 12321–12332. PMLR.

Zhang, Ke, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video summarization with long short-term memory. In *European conference on computer vision*, 766–782. Springer.

Zhang, Qingchen, Laurence T Yang, Xingang Liu, Zhikui Chen, and Peng Li. 2017. A tucker deep computation model for mobile multimedia feature learning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 13(3s):39.

Zhang, Weiyu, Praveen Srinivasan, and Jianbo Shi. 2011. Discriminative image warping with attribute flow. In *Computer vision and pattern recognition 2011*, 2393–2400. IEEE.

Zhang, Ye, and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

Zhang, Zhengwu, Jingyong Su, Eric Klassen, Huiling Le, and Anuj Srivastava. 2018. Rate-invariant analysis of covariance trajectories. *Journal of Mathematical Imaging and Vision* 1–18.

Zhen, Xingjian, Rudrasis Chakraborty, Nicholas Vogt, Barbara B. Bendlin, and Vikas Singh. 2019. Dilated convolutional neural networks for sequential manifold-valued data. In *International conference on computer vision*.

Zheng, Liang, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. 2015. Scalable person re-identification: A benchmark. In *Proceedings of the ieee international conference on computer vision*, 1116–1124.

Zhou, Bolei, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 2921–2929.

Zhou, Bolei, Yiyou Sun, David Bau, and Antonio Torralba. 2018. Interpretable basis decomposition for visual explanation. In *Proceedings of the european conference on computer vision (eccv)*.

Zhou, Kaiyang, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. 2019. Omni-scale feature learning for person re-identification. In *Proceedings of the ieee/cvf international conference on computer vision*, 3702–3712.

Zhou, S., J. Lafferty, et al. 2010. Time varying undirected graphs. *ML* 80(2-3):295–319.

Zhu, Hongtu, Yasheng Chen, Joseph G Ibrahim, Yimei Li, Colin Hall, and Weili Lin. 2009. Intrinsic regression models for positive-definite matrices with applications to diffusion tensor imaging. *Journal of the American Statistical Association* 104(487): 1203–1212.