Model Drift in Brain-Computer Interfaces and Space-Based Remote Sensing

by

Mona Khoshnevis

B.Sc., Mechanical Engineering, Amirkabir University of Technology, 2013

B.Sc., Pure Mathematics, Amirkabir University of Technology, 2014

M.Sc., Pure Mathematics, Sharif University of Technology, 2017

A dissertation submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in the Division of Applied Mathematics at Brown University

PROVIDENCE, RHODE ISLAND

October 2024

This dissertation by Mona Khoshnevis is accepted in its present form by the Division of Applied Mathematics as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date _____ _____

Matthew Harrison, Ph.D., Advisor

Recommended to the Graduate Council

Date _____ _____

Björn Sandstede, Ph.D., Reader

Date _____ _____

John Simeral, Ph.D., Reader

Date _____ _____

James Kellner, Ph.D., Reader

Approved by the Graduate Council

Date _____ _____

Thomas A. Lewis, Dean of the Graduate School

Curriculum Vitae

Mona Khoshnevis graduated from Amirkabir University of Technology (Tehran Polytechnic) where she received a B.Sc. in Mechanical Engineering and Mathematics as a dual degree. She also graduated from Sharif University of Technology with a M.Sc. in Pure Mathematics.

Dedication

To my mother, Maliheh (Sorayya) Alesheikh.

Acknowledgements

When I started graduate school at Brown, the first class I attended was Information Theory, taught by Professor Matthew Harrison. I vividly recall how amazed I was as I watched him set the stage for a semester of learning. Later on, each lecture note, every meticulously crafted homework solution, all his thoughtful responses to students' questions, and the elegance of his teaching style left an indelible impression on me. I had been fortunate to learn from many remarkable instructors throughout my undergraduate and master's program, yet Professor Harrison's approach stood apart—unmatched in every respect. Regrettably, during that time, I was navigating through a storm of personal challenges, including the profound pain of grief, which hindered my ability to fully immerse myself in that invaluable class, but now, looking back from a better place, I know I would do my best to teach like him if I ever lead an undergraduate or graduate math course. I believe he is the best mathematics instructor one can aspire to be.

Later in my PhD, I had the honor of being Professor Harrison's graduate teaching assistant. This role allowed me to witness his teaching style up close, and I was truly astonished. I learned so much from his approach to teaching, from grading to exam preparation and course design. His dedication and mastery have been an incredible source of inspiration for me.

During my third year, I had the greatest honor and privilege of my life: being accepted by Professor Harrison as his PhD student. As my research journey began, I found myself amazed daily by how someone could be the best advisor as well as the best teacher.

He was the perfect mentor—supportive, encouraging, understanding, and endlessly patient.

Thank you, Dr. Harrison. Words cannot fully capture how profoundly grateful I am for everything I have learned from you, both academically and personally. Thank you for your patience in explaining the details in every meeting, for the time that you invested in

my growth, and for guiding me onto the best academic path, even when I had no hope. Beyond research, thank you for always giving me the vision to see things from a deeper perspective, and for illuminating my way so I wouldn't feel lost.

I have always searched for a role model in life, and I am deeply thankful to you, Professor Harrison, for being the best role model I could have ever asked for.

I am very grateful to Professor Bjorn Sandstede, Professor John Simeral, and Professor James Kellner for serving on my thesis committee. Thank you for taking the time to read my dissertation and for your invaluable feedback and guidance.

I would also like to express my appreciation to my fellow graduate students, Ewina Pun and Dafeng Zhang, for their collaboration and for always being there to answer my questions.

To my dearest friends Fatemeh, Emad, Maryam, Behzad, and Maryam—thank you for the endless conversations, joy, and laughter we shared. Your friendship has been a bright light during these years.

I am eternally grateful to my parents: Mohsen Khoshnevis and Maliheh (Sorayya) Alesheikh. I wish my father could be here today, even more than on every day I have missed him during these years. I take comfort in knowing he is the happiest man in heaven now. I deeply thank my mother for her selfless support, kindness, and care. Without her unwavering support, none of this would have been possible.

And finally, thank you, Mohsen. No words can truly express how grateful I am to you, every single minute, from the depths of my heart. I am incredibly lucky to have you by my side.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

In real-world applications, predictive machine learning models often exhibit different performance at test time compared to their original training time. This discrepancy can arise from various factors, but in this study, we assume that the models for training and testing are temporally independent and have changed over time. Specifically, we model this phenomenon as a shift in the joint distribution between the target and explanatory variables from training to testing.

This phenomenon is referred to by various terms in the literature, including concept shift/drift [1–5], dataset shift/drift [6–8], and model drift [9, 10]. For the purposes of this study, we will adopt the term "model drift", acknowledging that its usage may vary across different contexts in the field. Letting X represent the feature random vector and Y the response variable, model drift accounts for performance degradation by assuming that the joint distribution $p(X, Y)$ differs between training and testing. This change in the joint distribution can arise from various sources: a shift in the posterior probability distribution $p(Y|X)$, a shift in the input or feature data probability distribution $p(X)$ (covariate shift [6]), a shift in the prior probability distribution of the target labels $p(Y)$, a shift in the conditional probability distribution $p(X|Y)$, or a combination of the first two or the last two, as $p(X, Y) = p(X)p(Y|X) = p(Y)p(X|Y)$. Depending on the application, we may focus on one of these aspects. Some of the terms mentioned earlier

are often reserved by certain authors for specific types of changes; for example, some refer to "concept shift" exclusively when $p(Y|X)$ changes [6, 11, 12].

Drift detection is a fundamental component of research on model drift [13, 14]. A common approach involves comparing data from two different time periods. This typically entails assuming a probabilistic model for the data and applying a hypothesis test to assess the statistical significance of any changes between the models. The focus of the first chapter of this work is to detect model drift in a real-world research field: brain-computer interfaces.

Brain-Computer Interfaces (BCIs) translate neural signals into commands for assistive technologies, primarily benefiting individuals with paralysis. High-performance BCIs rely on neural decoders calibrated during tasks like target-acquisition, where users are instructed to reach specific targets on a screen. However, these decoders degrade over time due to changes in the relationship between neural signals and intended movements [10, 15–25]. Therefore, to enhance performance, it is crucial to address the underlying non-stationarity, i.e., the model drift in the joint distribution of neural signals and kinematic intentions. In BCI literature, it is common to model the joint distribution of the neural activity (X) and the intended kinematics features (Y) via $p(Y)$ and $p(X|Y)$, and then use Bayes rule to derive a predictive model $p(Y|X)$ [10, 26–30]. The first chapter focuses on developing two p-values to detect model drift in the conditional probability distribution of neural signals given intended movements ($p(X|Y)$) under a linear Gaussian encoding model. To this end, the first chapter designs and analyzes two p-values to detect model drift, specifically to assess statistically significant changes in $p(X|Y)$ under a linear Gaussian encoding model. The method is explicitly designed to be invariant to changes in $p(Y)$, which is important in the application.

However, in real-world BCI applications, the primary concern shifts from only detecting whether model drift has occurred, as discussed in the first chapter, to understanding the severity of the drift—specifically, by quantifying how similar the new joint distribution $p(X, Y)$ is to the previous one [13]. Distribution-based drift detection methods

can effectively measure the severity of model drift. These algorithms work by using a distance function or metric to assess the dissimilarity between the distributions of the two epochs of data [13]. Dasu et al. [31] employed an Information-Theoretic Approach (ITA) for drift detection, utilizing relative entropy, also known as Kullback-Leibler (KL) distance [32], to measure the distribution shift between two sliding windows in a multi-dimensional data stream, followed by bootstrap methods to establish the statistical significance of these distances. Pun et al. [10] conducted a similar approach, using Gaussian Kullback-Leibler distance (GKLD) between the sliding windows of Principal Component Analysis (PCA) transformations of neural data recorded from two BrainGate participants. They demonstrated that GKLD strongly correlates with the performance of the initially calibrated decoder. Since their method for addressing model drift focuses on the input data probability distribution $p(X)$, their experimental results suggest that performance degradation caused by model drift can be largely explained by differences in the marginal distributions of neural activity ($p(X)$). In the second chapter of this work, we further explore this by asking: 1) What is the relationship between the GKL distance of $p(X)$ and the different components of error? Is it always linear under various model drift paradigms? 2) Can the high correlation that is observed in practice be a result of the model noise itself, or is it solely due to model drift?

With the development of advanced Brain-Computer Interface (BCI) systems capable of decoding multiple tasks, such as point-and-click or click-and-drag interfaces [33–39], a new type of performance degradation has emerged. This degradation presumably occurs due to interference between the primary task decoder and secondary task intentions, resulting in unintended actions like accidental cursor movement during gesturing [28, 35, 40]. We model this phenomenon as an example of model drift, where a system trained on single-task data (e.g., movement) is later required to decode multiple tasks (e.g., cursor movement and clicking). This scenario can be viewed as a covariate shift, where $p(X)$ changes due to alterations in the neural data, which is assumed to be affected by the person's motor intentions. Addressing this issue requires a novel approach, distinct from multi-label or multi-output learning [41], as training data typically includes separate

datasets for different tasks. The third chapter explores two strategies to mitigate this type of model drift. The first involves a data-dependent regularization technique to penalize dependencies between the primary task decoder and interfering tasks while maintaining decoding accuracy. The second approach identifies a neural data subspace where the primary task linear decoder is robust to any "anticipated" model drift, caused by potential variations in the user's secondary task intention.

In the final chapter, a different type of model change between training and test phases is addressed, primarily involving measurement error [42, 43]. Specifically, in covariate measurement error, the true value of the covariate variable in the model may differ from what is obtained through the data collection process. For instance, while the response variable Y at test time needs to be predicted based on X, X is not observed during training. Instead, a surrogate measurement X' is available along with Y. This scenario is exemplified in NASA's Global Ecosystem Dynamics Investigation (GEDI) research, where the goal is to predict aboveground biomass density (AGBD) from spaceborne lidar waveform features at the footprint level [44, 45]. However, since datasets with actual on-orbit GEDI measurements (X) paired with AGBD (Y) are limited, model calibration becomes challenging [45, 46]. Due to the limited availability of such training data, simulated waveforms from airborne lidar (X') are used as a surrogate, resulting in a training dataset with AGBD (Y) paired with these simulated waveform features, i.e., sample pairs of (X', Y) [47, 48]. Additionally, a dataset with presumably co-located sample pairs of (X', X) exists. Given the absence of a dataset containing all three variables (Y, X, X'), the challenge is to determine the best method to predict Y from X using only the training dataset of (X', Y) and the supplementary dataset of (X', X). To address this challenge, we propose two models: The first model leverages a conditional independence assumption where Y is conditionally independent of X given X', i.e., $p(Y|X', X) = p(Y|X')$, and uses a sequential estimation approach to estimate $\mathbb{E}[Y|X]$, demonstrating improved performance over traditional methods that estimate $\mathbb{E}[Y|X']$. The second model incorporates a regularization term to penalize inconsistencies in the predictor map based on the (X', X) dataset.

CHAPTER 2

A multiple hypothesis test to detect encoding model drift in brain-computer interfaces

2.1 Introduction

Brain-computer interfaces (BCIs) hold significant promise for translating neural population recordings into motor imagery actions, in both able-bodied non-human primates [26, 49–54] and in tetraplegic humans [29, 38, 55–57]. This translation, known as decoding, relies on static functional mapping between neural activities and motor imagery parameters [15, 24, 58–60]. The decoder is learned through a calibration process, which generates pairs of neural signals and motor imagery action signals (e.g., cursor velocity) through instructed targets. However, this mapping may not remain static over time, as shown in [15–25]. If a decoder trained on the initial mapping is continuously used, its performance can deteriorate significantly. Studies have shown that the performance of trained decoders indeed degrades over time due to various factors, including temporal changes in neural tuning properties [61–64], plasticity [65–69], and technical limitations of the recording device [22, 70, 71]. Therefore, it is crucial to

identify when the functional mapping has changed to address the potential need for retraining the decoding model.

The functional mapping between the neural activity and motor intentions is modeled by linear Gaussian model widely in the field [24, 26, 28–30]. To determine any statistically significant alteration of this model, Kim et al. [24] developed a statistical methodology to compare these linear Gaussian probabilistic encoding models between two epochs. Their method involves measuring the Bhattacharyya distance and Kullback-Liebler (KL) divergence between the likelihood of neural signals given identical distributions of a two-dimensional movement intention. They then use a hypothesis test to identify if the distance is statistically significant.

In contrast to [24], we propose two hypothesis tests with the null hypothesis that the two probabilistic encoding models are the same. Our proposed tests are built on some classical results for univariate response variable described in [72–74]. This approach avoids the assumption of identical kinematics distributions, which is addressed through a separate statistical test in their work. We further solidify this by demonstrating the validity of both tests in both scenarios: when the motor imagery variable distributions are the same and when they differ. This validation is performed using synthetic datasets.

Furthermore, the model drift we identify here is not limited to changes in individual neuron directional tuning, as seen in [63, 68, 75, 76]. Instead, it captures any statistically significant variation in the entire encoding model which represents the ensemble response distribution given the behavioral variables.

To assess the power of both tests under the alternative hypothesis, we employed various quantitative model perturbations using simulation. These perturbations encompass all different cases suggested in [17], including changes in the mean firing rate, preferred direction, noise, modulation depth, and channel dropout.

The proposed multiple hypothesis test is then applied to two real-world data comprising pairs of neural signals and movement intentions collected from two individuals with tetraplegia (T11 and T5) during a computer cursor control task using an iBCI. Both

individuals performed target acquisition tasks on a screen, allowing inference of their motor imagery intentions. This enabled us to collect the necessary samples for applying our hypothesis test.

The results suggest statistically significant model drifts in almost all session pairs, except for consecutive sessions, for both participants. These findings are detailed in the supplementary material (Supplementary Figure 3) of Pun et al. [10]. Additionally, the results align with observations of model drift in non-human primates reported by Kim et al. [24].

2.2 Hypothesis test

Consider the multivariate regression model of X given $Y = y$ described by

$$X = yA + b + Z \tag{2.1}$$

where $X, b, Z \in \mathbb{R}^{1\times d}$, $y \in \mathbb{R}^{1\times \ell}$, $A \in \mathbb{R}^{\ell\times d}$, and $Z \sim N_d(0, \Sigma)$. The unknown parameters are $\theta = (A, b, \Sigma)$. Let $(y_1, X_1), \ldots, (y_n, X_n)$ be a collection of observations from this regression model, where we assume that the corresponding $Z_1, \ldots, Z_n$ are independent and unobserved. Let $k = \ell + 1$. Concatenate the data into the $n \times k$ design matrix $\mathbf{Y}$ and the $n \times d$ response matrix $\mathbf{X}$. The j-th row of $\mathbf{X}$ is X_j and the j-th row of $\mathbf{Y}$ is $(1\ y_j)$. The classical unbiased estimators for $\beta = \left(\begin{smallmatrix} b \\ A \end{smallmatrix}\right)$ and Σ are

$$\widehat{\beta} = (\mathbf{Y}^\top \mathbf{Y})^{-1} \mathbf{Y}^\top \mathbf{X}$$

and

$$\widehat{\Sigma} = \frac{\widehat{\mathbf{R}}}{n-k} \qquad \text{for} \quad \widehat{\mathbf{R}} = (\mathbf{X} - \mathbf{Y}\hat{\beta})^\top (\mathbf{X} - \mathbf{Y}\hat{\beta}).$$

These are defined whenever $\mathbf{Y}^\top \mathbf{Y}$ is invertible and $n > k$.

Suppose that we have two independent datasets[1] from this model with possibly

[1] by independent datasets, we mean $(Z'_1, \ldots, Z'_{n'})$ is independent of $(Z''_1, \ldots, Z''_{n''})$, in fact, we mean conditional independence of X' and X'' given Y' and Y''.

different parameters:
$(y'_1, X'_1), \ldots, (y'_{n'}, X'_{n'})$ with parameters $\theta' = (A', b', \Sigma')$ and $(y''_1, X''_1), \ldots, (y''_{n''}, X''_{n''})$ with parameters $\theta'' = (A'', b'', \Sigma'')$. We want to test the null hypothesis

$$H_0 : \theta' = \theta''.$$

In the following, we will give two different valid p-values for H_0. Recall that a p-value pv is *valid* if $\mathbb{P}(\mathrm{pv} \leq \alpha) \leq \alpha$ for all $\alpha \in [0, 1]$. The first, pv_0, is only valid under the simplified model that $\Sigma = cI_d$, a constant multiple of the identity matrix. The second, pv_1, is valid for arbitrary Σ, but has the disadvantage of not having power to detect $\Sigma'_{ij} \neq \Sigma''_{ij}$ for $i \neq j$.

Combine the two independent datasets into a single dataset $(y_1, X_1), \ldots, (y_n, X_n)$ for $n = n' + n''$. Construct $\hat{\beta}$, $\widehat{\mathbf{R}}$, and $\widehat{\Sigma}$ as defined above. By analogy, construct $\widehat{\beta}'$, $\widehat{\mathbf{R}}'$, $\widehat{\Sigma}'$ and $\widehat{\beta}''$, $\widehat{\mathbf{R}}''$, $\widehat{\Sigma}''$ for the original datasets $(y'_1, X'_1), \ldots, (y'_{n'}, X'_{n'})$ and $(y''_1, X''_1), \ldots, (y''_{n''}, X''_{n''})$, respectively. We assume that all of these estimators are well-defined throughout, which places restrictions on the amount of data and the structure of the design matrices. Let $F_{d_1,d_2}(y)$ denote the cumulative distribution function of an F-distributed random variable with degrees of freedom d_1 and d_2, and let $G_{d_1,d_2} = 1 - F_{d_1,d_2}$ be the corresponding right tail probability. Fix positive constants λ_i and λ_{ij} such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and $\sum_{i=1}^{3} \sum_{j=1}^{d} \lambda_{ij} = 1$. In the experiments below we use $(\lambda_1, \lambda_2, \lambda_3) = (1/4, 1/4, 1/2)$, $\lambda_{1j} = 1/(4d)$, $\lambda_{2j} = 1/(4d)$, and $\lambda_{3j} = 1/(2d)$.

Define

$$T_\Sigma = \frac{n'-k}{n''-k} \cdot \frac{\mathrm{Tr}(\widehat{\mathbf{R}}'')}{\mathrm{Tr}(\widehat{\mathbf{R}}')}$$

$$T_\beta = \frac{n-2k}{k} \cdot \frac{\mathrm{Tr}(\widehat{\mathbf{R}}) - \mathrm{Tr}(\widehat{\mathbf{R}}') - \mathrm{Tr}(\widehat{\mathbf{R}}'')}{\mathrm{Tr}(\widehat{\mathbf{R}}') + \mathrm{Tr}(\widehat{\mathbf{R}}'')}$$

$$\mathrm{pv}_0 = \min\left\{\lambda_1^{-1} F_{d(n''-k),d(n'-k)}(T_\Sigma),\ \lambda_2^{-1} G_{d(n''-k),d(n'-k)}(T_\Sigma),\ \lambda_3^{-1} G_{dk,d(n-2k)}(T_\beta),\ 1\right\}$$

$$T_{\Sigma,j} = \frac{n'-k}{n''-k} \cdot \frac{\widehat{\mathbf{R}}''_{jj}}{\widehat{\mathbf{R}}'_{jj}} \quad (j = 1, \ldots, d)$$

$$T_{\beta,j} = \frac{n-2k}{k} \cdot \frac{\widehat{\mathbf{R}}_{jj} - \widehat{\mathbf{R}}'_{jj} - \widehat{\mathbf{R}}''_{jj}}{\widehat{\mathbf{R}}'_{jj} + \widehat{\mathbf{R}}''_{jj}} \quad (j = 1, \ldots, d)$$

$$\mathrm{pv}_1 = \min_{j=1,\ldots,d} \min\left\{\lambda_{1j}^{-1} F_{n''-k,n'-k}(T_{\Sigma,j}),\ \lambda_{2j}^{-1} G_{n''-k,n'-k}(T_{\Sigma,j}),\ \lambda_{3j}^{-1} G_{k,n-2k}(T_{\beta,j}),\ 1\right\}$$

Theorem 1. *Under the above modeling assumptions,* pv_1 *is a valid p-value for* H_0 *whenever it is well-defined. Under the additional modeling assumption that* $\Sigma = cI_d$, pv_0 *is a valid p-value for* H_0 *whenever it is well-defined.*

Proofs are in the Appendix and rely heavily on the classical results described, for example, in [72] for the special case of $d = 1$ (univariate response).

2.3 Numerical experiments

In this section, we corroborate our suggested p-values in Theorem 1 on some examples of both synthetic and real data.

2.3.1 Synthetic data

2.3.1.1 Setup

In order to check out the proposed tests performance, we synthesize two datasets $(y'_1, X'_1), \ldots, (y'_{n'}, X'_{n'})$ and $(y''_1, X''_1), \ldots, (y''_{n''}, X''_{n''})$ in the following way.

First, y'_i's are drawn from multivariate Gaussian distributions $N_l(m', \Gamma')$ and y''_i's from $N_l(m'', \Gamma'')$, where l is the dimension of y vectors. The same datasets of Y' and

Y'' have been used throughout the study.

Now to synthesize X's for each example, given $Y'_i = y'_i$ ($Y''_i = y''_i$), X'_i (X''_i) is drawn from the multivariate regression model with parameters θ' (θ'') as explained in section 2.2 (Note that to test pv_0, we set an additional assumption for θ' and θ'' that Σ' and Σ'' are both constant multiples of identical matrix).

In section 2.3.1.2, we check the power of the tests by finding the empirical probability of rejection while varying alternatives. Then, in section 2.3.1.3, we depict the tests power for a fixed perturbation level, by varying the sample size. Finally, in section 2.3.1.4, we argue that pv_0 is not a p-value when the assumption of same and independent Gaussian noise for each response dimension is not met. The number of Monte-Carlo simulation to find the empirical probability of rejection we used for every case is M.

Throughout, we set $l = 2$, $M = 1000$ and the significance level is 0.05.

2.3.1.2 Power of the tests for perturbing alternative

To verify validity of both tests pv_0 and pv_1, as well as to check their performance (type II error) when the null does not hold, we create varying alternative hypothesis, i.e., we perturb θ'' from θ' by a defined perturbation measure. In every figure throughout this section, each point represents a specific perturbation, whose level is captured in x-axis, while the y-axis shows the empirical power of the test in that perturbation level, using significance level of 0.05.

Also, the left panel in every figure is for the same distribution of Y' and Y'', where we set $m' = m'' = \mathbf{0}$ and $\Gamma' = \Gamma'' = I$. We also have $n' = 53$ and $n'' = 47$.[2]

While for the right panel, distribution of Y' and Y'' is different: $m' = [0, 0], m'' = [10, 10]$, and $\Gamma' = I, \Gamma'' = 3I$. Plus, $n' = 52$ and $n'' = 48$.

Also, wherever the parameter used for perturbation is t, we are using 100 equally distanced values in $[-0.5, 0.5]$ for t.

[2] n' was drawn from $Binomial(n, \frac{1}{2})$ where $n = 100$, and $n'' = n - n'$.

1. pv_1:

   Here, we show the empirical power of the test pv_1 by varying the perturbation level. Different paradigms of perturbation are captured in the following cases:

   (I) **Change of regression coefficients** A' while other parameters b' and Σ' are fixed. This includes scaling, rotation, and an additional scenario in which in which A'' is chosen randomly in a varying Frobenius distance from A'. Scaling is showed in Figure 2.1 for $d = 1$, and in Figure 2.2 for $d = 4$. Rotation is covered in Figure 2.3. The last scenario is examined in Figure 2.4 for 1-dimensional response case, and then in multidimensional cases of figures 2.5 and 2.6 ($d = 4$ in both), two different fashions of choosing A'' in total Frobenius distance from A' are presented. For description of each of these methods in detail, refer to the figure captions. It is worth noting that, as mentioned in section 1, A is the last l rows of β, and b is the first row.
   Perturbation measure for all analysis of this case is the Frobenius distance between A' and A'' ($||A' - A''||_F$).

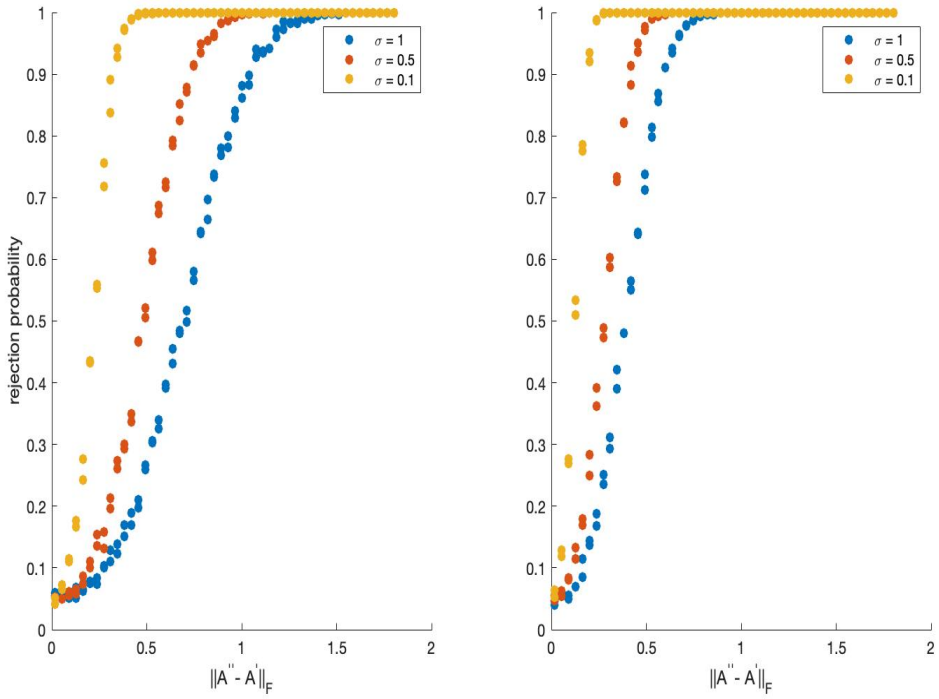


Figure 2.1: pv_1 power for change of regression coefficients - 1 dimensional response: we set $\beta' = [1, 2, 3]^\top$ (i.e., $A' = [2, 3]^\top$ and $b' = 1$). We also consider three different cases for the variance $\Sigma' = \Sigma'' \in \{0.1, 0.5, 1\}$, and we vary A as follows: $A'' = (1 + t)A'$ while $b'' = b' = 1$.

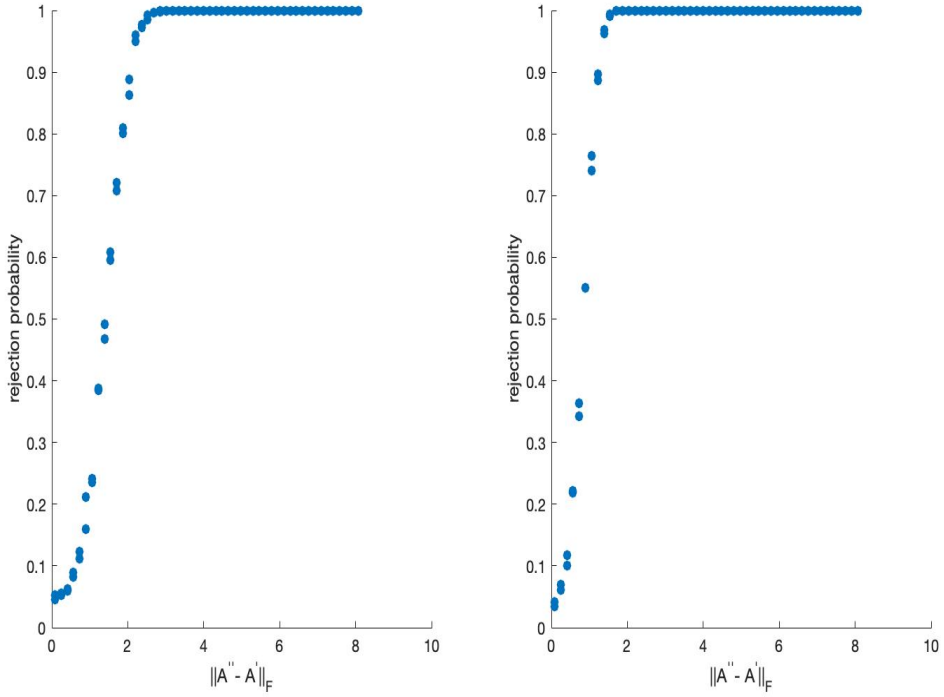


Figure 2.2: pv_1 power for change of regression coefficients - multidimensional response: Here, $\beta' = \begin{bmatrix} 1 & 3 & 10 & 0 \\ 3 & 5 & 6 & 8 \\ 10 & 0.5 & 5 & 1 \end{bmatrix}$, $\Sigma' = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 5 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 3 \end{bmatrix}$, and $A'' = (1+t)A'$ while Σ and b are fixed ($b' = b'' = \begin{bmatrix} 1 & 3 & 10 & 0 \end{bmatrix}$).

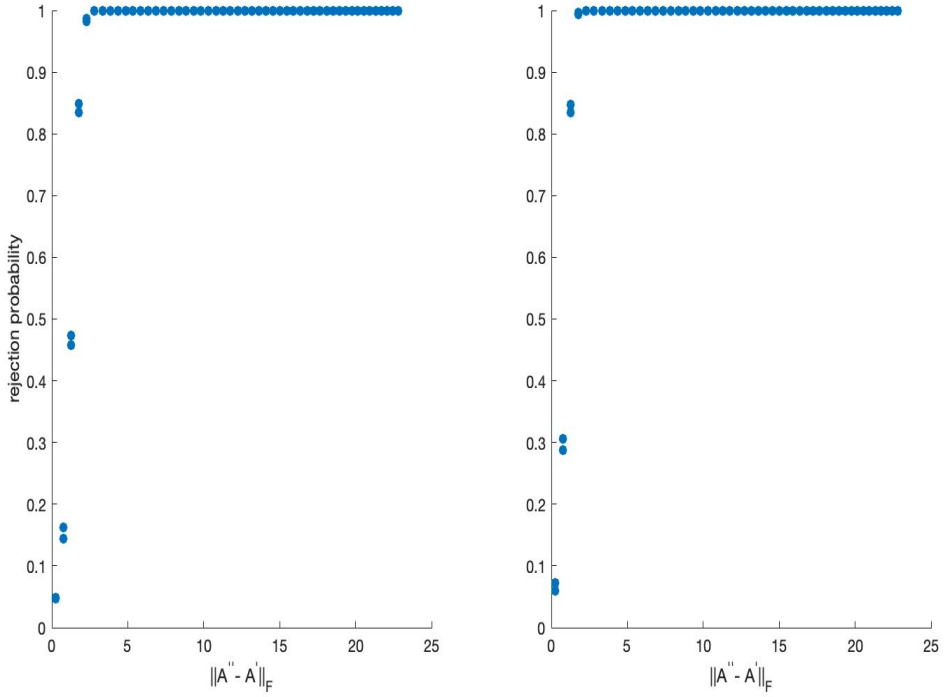


Figure 2.3: pv_1 power for rotation of A - multidimensional: we have $\beta' = \begin{bmatrix} 1 & 3 & 10 & 0 \\ 3 & 5 & 6 & 8 \\ 10 & 0.5 & 5 & 1 \end{bmatrix}$, $\Sigma' = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 5 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 3 \end{bmatrix}$, and A'' is the rotated version of A', i.e., $A'' = \begin{bmatrix} \cos(t\pi) & -\sin(t\pi) \\ \sin(t\pi) & \cos(t\pi) \end{bmatrix} A'$, while Σ and b remain unchanged.

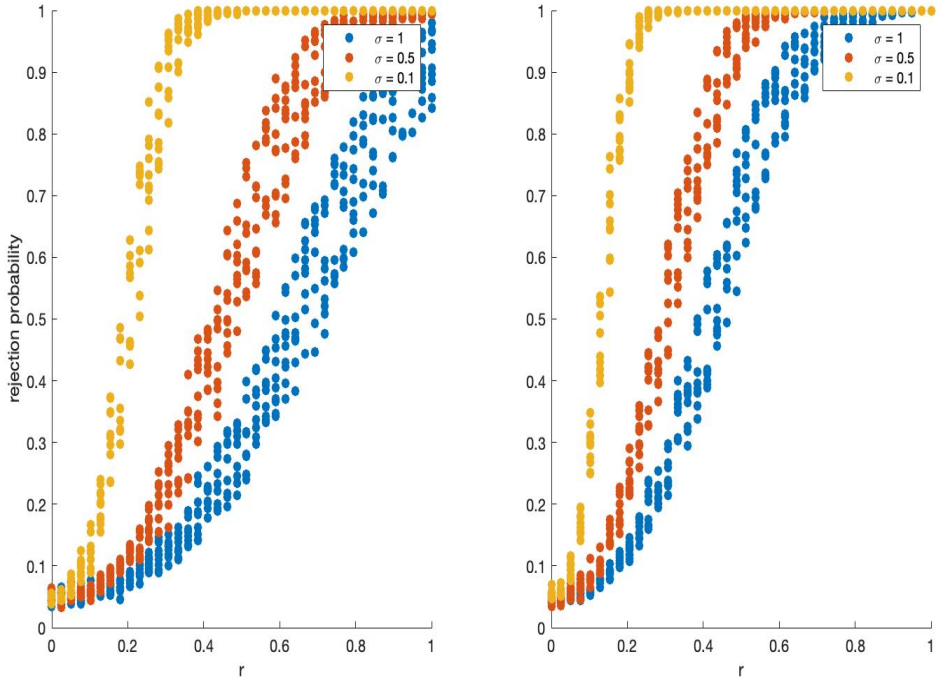


Figure 2.4: pv_1 power when A'' is randomly chosen in a distance r from A' - 1 dimensional response: As in Figure 2.1, we consider three different cases for the variance $\Sigma' = \Sigma'' \in \{0.1, 0.5, 1\}$, and we have $\beta' = [1, 2, 3]^\top$. Then for each alternative, with fixed Σ and b, we create 10 different randomly chosen A'' on a sphere centered at A' with radius r, where r takes 40 equally distanced values from 0 to 1.

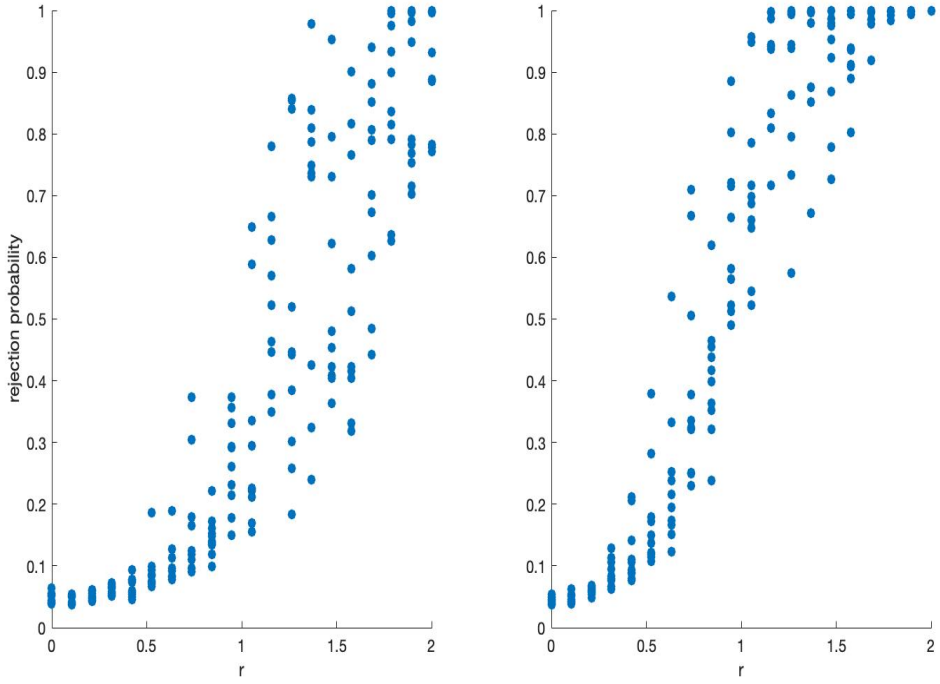


Figure 2.5: pv_1 power when A'' is randomly chosen in a distance r from A' - multidimensional response: we have $\beta' = \begin{bmatrix} 1 & 3 & 10 & 0 \\ 3 & 5 & 6 & 8 \\ 10 & 0.5 & 5 & 1 \end{bmatrix}$, and $\Sigma' = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 5 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 3 \end{bmatrix}$, and A'' takes 10 randomly chosen matrices of Frobenius distance r from A', while Σ and b are being kept fixed (r take 20 equally distanced values between 0 and 2).

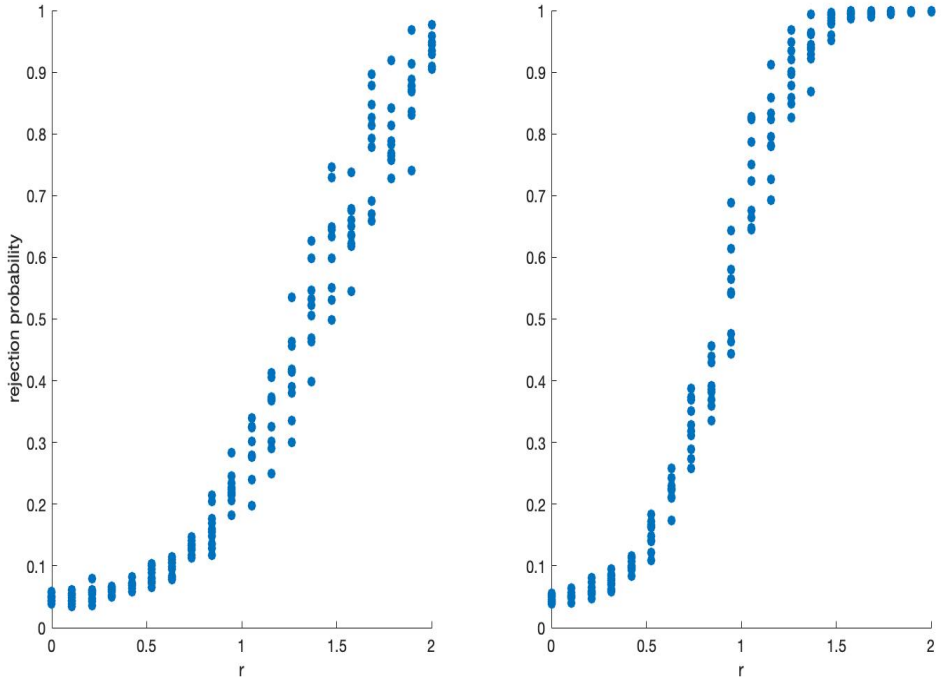


Figure 2.6: pv_1 power when A'' is randomly chosen in a Frobenius distance r from A' - multidimensional response: The setting is the same for Figure 2.5, with the difference that each column of A'' is located at the same distance from A', while the total Frobenius distance being kept as r.

(II) **Change of intercept** b', while A' and Σ being kept unchanged. This includes scaling of the intercept and is covered in Figure 2.7 for a case where $d = 4$. Perturbation measure is $||b' - b''||_F$.

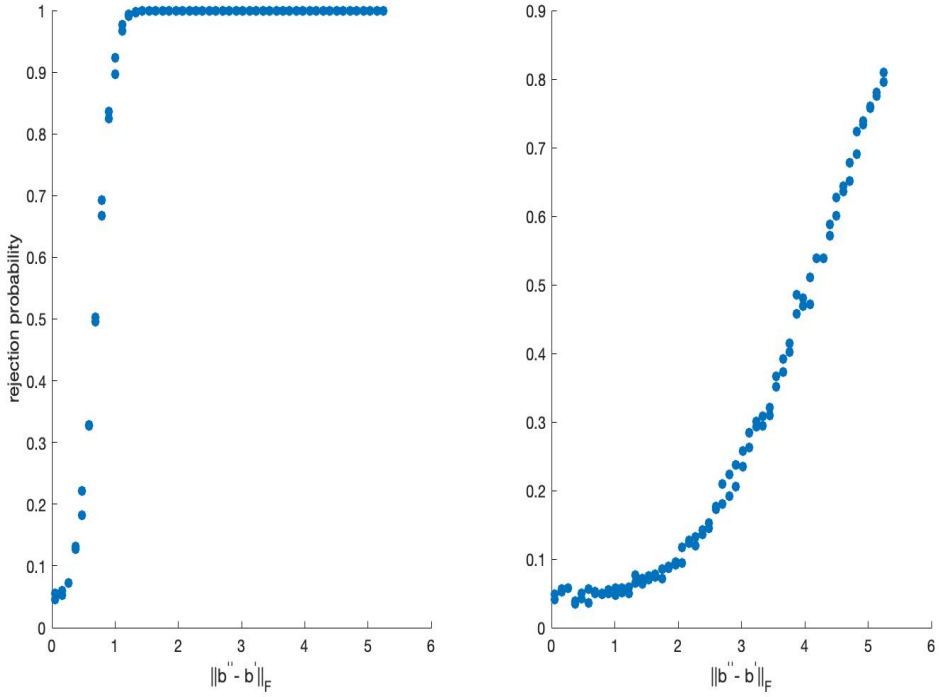


Figure 2.7: pv_1 power for change of intercept - multidimensional response: we have $\beta' = \begin{bmatrix} 1 & 3 & 10 & 0 \\ 3 & 5 & 6 & 8 \\ 10 & 0.5 & 5 & 1 \end{bmatrix}$, and $\Sigma' = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 5 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 3 \end{bmatrix}$, $\Sigma' = \Sigma''$, and $A' = A''$, but $b'' = (1+t)b''$.

(III) **Change of noise covariance parameter** Σ' while A' and b' remain unchanged. This is about scaling of the noise covariance matrix Σ', and it is represented in Figure 2.8 for $d = 1$ and in Figure 2.9 for $d = 4$.

Perturbation measure in this case is the geodesic distance below (See, e.g. [77]):

$$\delta_2(\Sigma', \Sigma'') = [\sum_{j=1}^{n} \log^2(\lambda_i(\Sigma'^{-1}\Sigma''))]^{1/2}$$

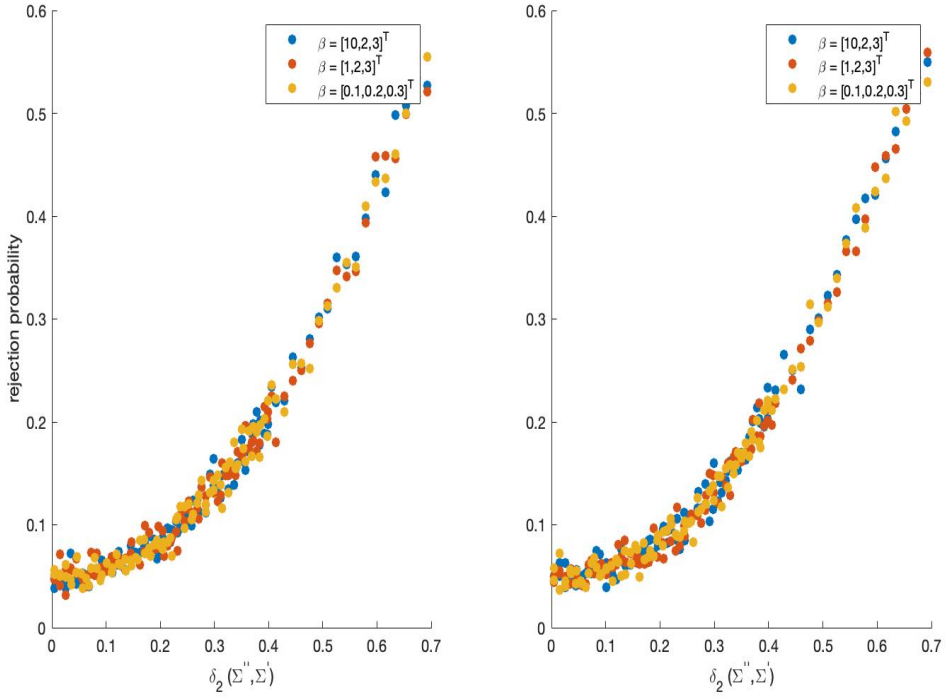


Figure 2.8: pv_1 power for change of noise - 1 dimensional response: we consider three different cases $\beta' = \beta'' \in \{[10,2,3]^\top, [1,2,3]^\top, [0.1,0.2,0.3]^\top\}$, changing Σ instead, where $\Sigma' = 1$, and $\Sigma'' = (1+t)\Sigma'$.

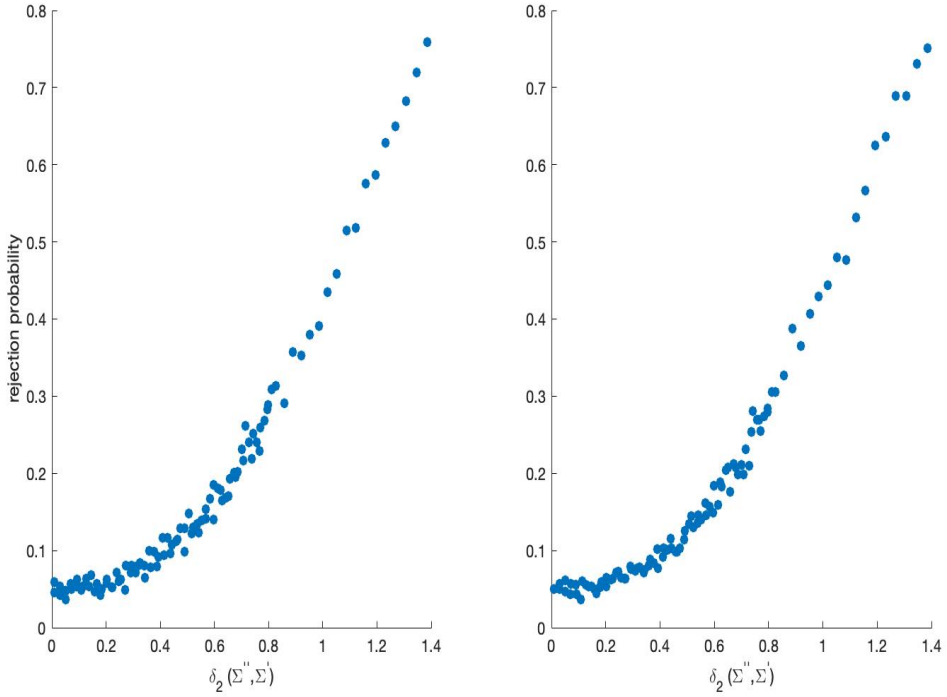


Figure 2.9: pv_1 power for change of noise - multidimensional response: we set $\beta' = \begin{bmatrix} 1 & 3 & 10 & 0 \\ 3 & 5 & 6 & 8 \\ 10 & 0.5 & 5 & 1 \end{bmatrix}$, $\Sigma' = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 5 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 3 \end{bmatrix}$, and $\Sigma'' = (1+t)\Sigma'$ while β is fixed.

(IV) **Sequential removal of both regression coefficients and intercept associated with each response dimension** while keeping Σ' term unchanged. The result of this analysis is covered in Figure 2.10, in which $d = 50$, and at each perturbation level i, we change the regression coefficients and the intercept associated with the first i dimensions to zero. See more details in the figure caption.

Perturbation level is i in this case.

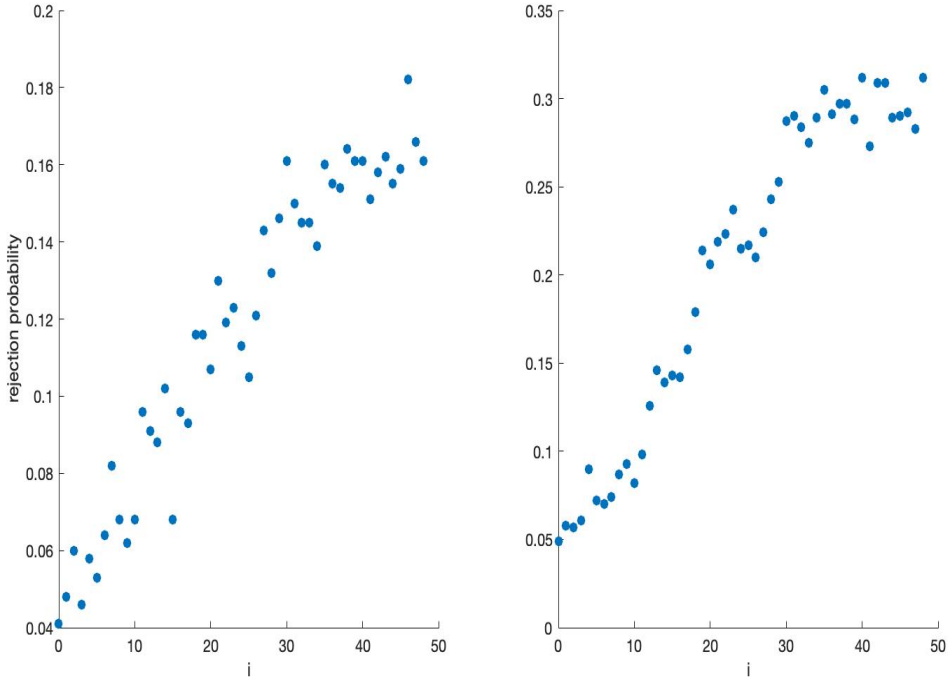


Figure 2.10: pv_1 power for dropping channels: we assume having a 50-dimensional ($d = 50$) response matrix X, so we choose a random $k \times 50$ matrix for β' and a random 50×50 symmetric positive-definite for Σ'. For the i'th alternative hypothesis ($i = 0, 1, 2, \ldots, 49$), we create β'' by changing the first i columns of β' to zero (Note that $i = 0$ means $\beta' = \beta''$).

2. pv_0:

   To investigate the performance of p-value pv_0, we provide a case of scaling regression coefficients while the other parameters are fixed in Figure 2.11, and a case of scaling noise covariance matrix while the other parameters remain unchanged in Figure 2.12. The perturbation level in the former case is $||A' - A''||_F$, and is $\delta_2(c', c'')$ in the latter, where $\Sigma' = c' I_d$ and $\Sigma'' = c'' I_d$.

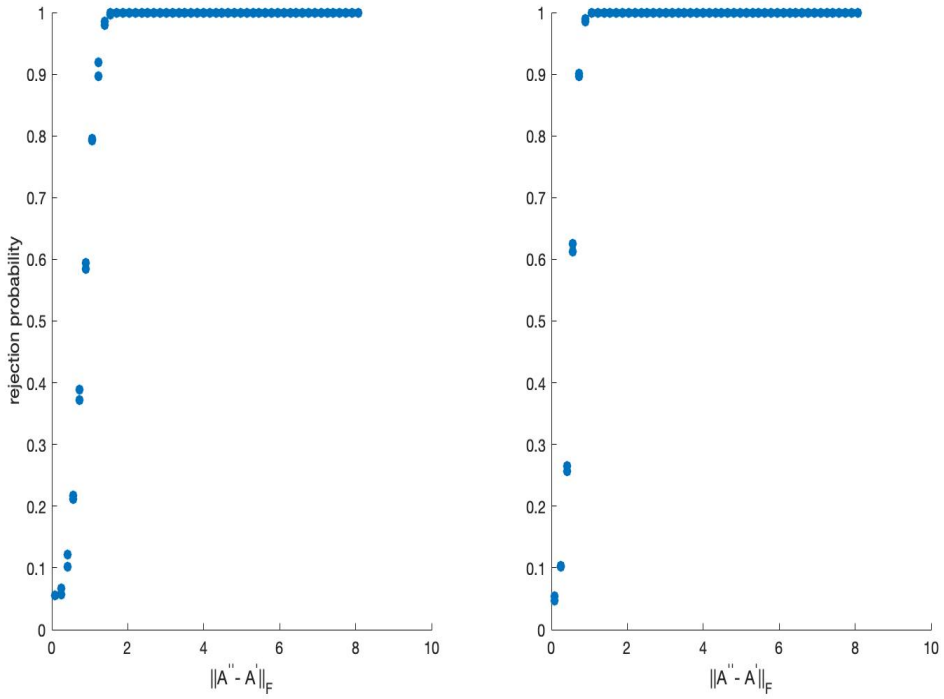


Figure 2.11: pv_0 power for change of regression coefficients - multidimensional response: we have $\Sigma' = c'I$ with $c' = 1$, $\beta' = \begin{bmatrix} 1 & 3 & 10 & 0 \\ 3 & 5 & 6 & 8 \\ 10 & 0.5 & 5 & 1 \end{bmatrix}$, and $A'' = (1+t)A'$ while $c'' = c'$ and $b'' = b'$.

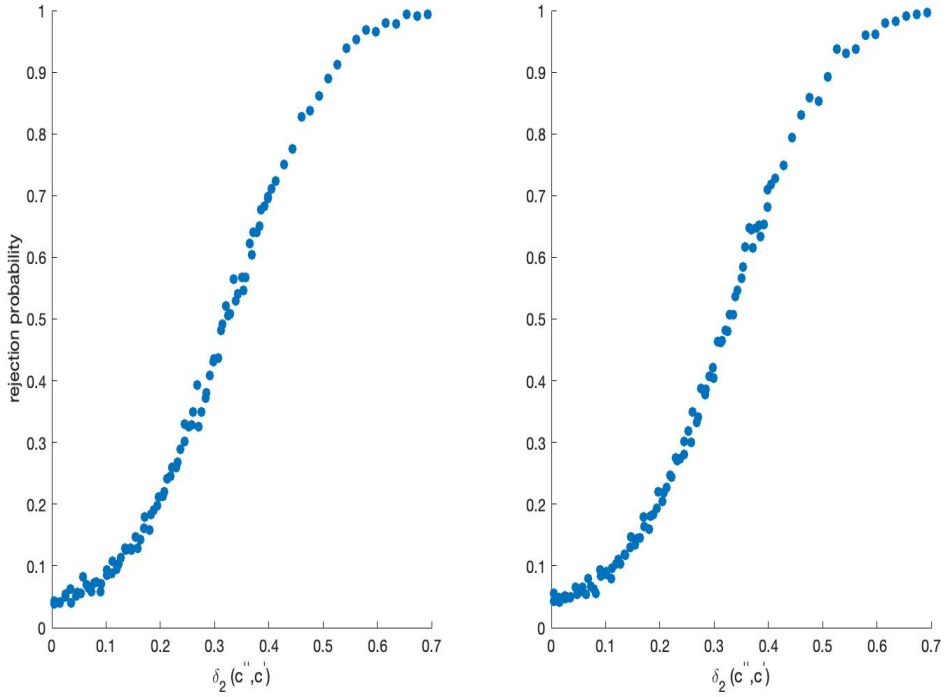


Figure 2.12: pv_0 power for change of noise covariance matrix - multidimensional response: we have $\Sigma' = c'I$ with $c' = 1$, $\beta' = \begin{bmatrix} 1 & 3 & 10 & 0 \\ 3 & 5 & 6 & 8 \\ 10 & 0.5 & 5 & 1 \end{bmatrix}$, and we change Σ as $c'' = (1+t)c'$ with fixed β.

In all figures above, we can observe that the empirical cumulative distribution function of both p-values pv_0 and pv_1 at 0.05 is less than 0.05 when we have zero perturbation, i.e., when the null hypothesis is true. This confirms that they are both valid p-values, which is analytically proved in the appendix. Plus, all figures above indicate that both tests are not too conservative, as they all start from close to our assigned significance level 0.05. Furthermore, all figures demonstrate that both methods have good power, as they converge to have zero empirical type II error as perturbation increases, so they are both capable of telling alternative and null hypothesis apart well. It is also worth noting that from figures 2.1 and 2.5, we can see that the lower the noise, the higher the power. Additionally, whether the distribution of Y' and Y'' are the same or different seems to have no effect on validity of both tests, while we see that it might have effects on the power. In fact, the more spread out Y' and Y'' are, the less type II error is achieved.

2.3.1.3 Power of the tests for changing sample size

In this section, we examine the power while we vary the sample size for both tests for a fixed perturbation level. Sample size n takes values in $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 125, 150, 17$ and the power of pv_1 and pv_0 is respectively shown in Figures 2.13 and 2.14 for these sample sizes. In both cases, $Y', Y'' \sim N_2(\mathbf{0}, I)$. The results confirm the known fact that the higher the sample size, the lower the type II error. See the figures caption for details.

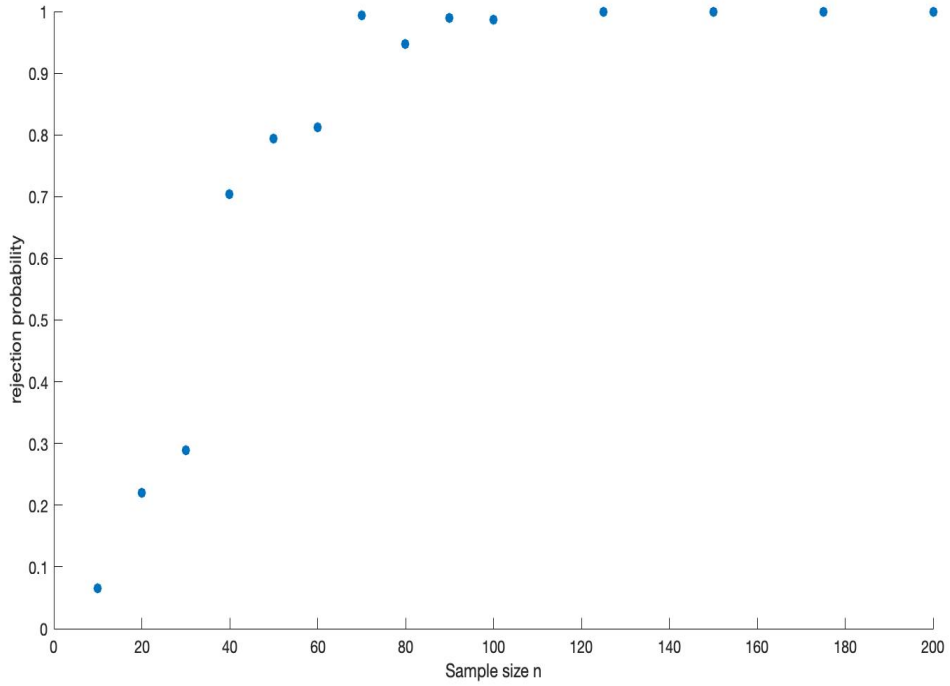


Figure 2.13: pv_1 power for change of sample size: we set $\beta' = \begin{bmatrix} 1 & 3 & 10 & 0 \\ 3 & 5 & 6 & 8 \\ 10 & 0.5 & 5 & 1 \end{bmatrix}$, $\Sigma' = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 5 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 3 \end{bmatrix}$, $\beta'' = 1.1\beta'$ and $\Sigma'' = 1.1\Sigma'$.

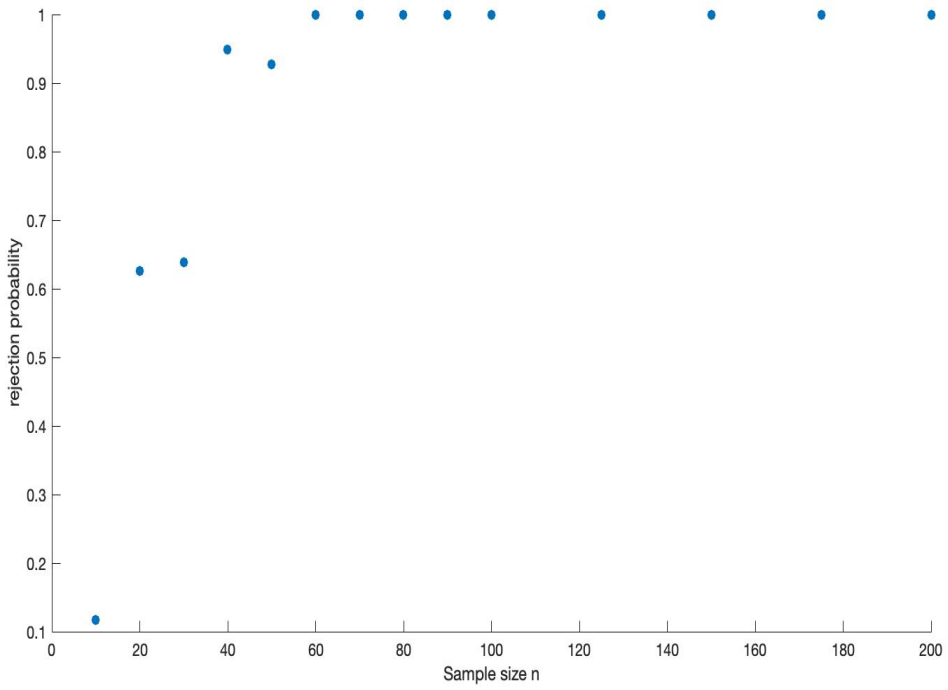


Figure 2.14: pv_0 power for change of sample size: $\beta' = \begin{bmatrix} 1 & 3 & 10 & 0 \\ 3 & 5 & 6 & 8 \\ 10 & 0.5 & 5 & 1 \end{bmatrix}$, $\Sigma' = c'I$ for $c' = 1$, $\beta'' = 1.1\beta'$ and $\Sigma'' = 1.1\Sigma'$.

2.3.1.4 Power of pv_0 for unmet conditions

Lastly, to address the question as to whether the p-value pv_0 is valid when the additional condition of $\Sigma = cI$ is not met, we examine the empirical rejection probability for the same significance level 0.05 by using pv_0 for a general model, where the assumption of $\Sigma = cI$ for the model in (2.1) does not hold. For the same setting of the left panels of figures 2.2 (2.9), we show the performance of pv_0 in figures 2.15 (2.16) below, and they show that pv_0 is not a p-value when the assumption of $\Sigma = cI$ is not met, since for zero perturbation, the empirical cdf of pv_0 is not less than or equal to the significance level anymore, which emphasizes the importance of using pv_1 for the general case.

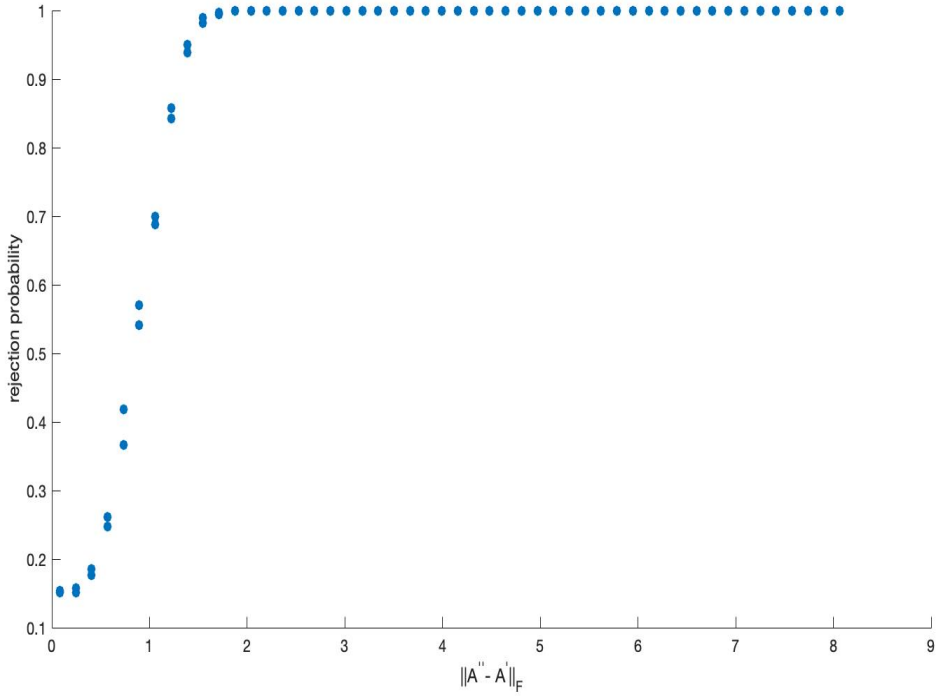


Figure 2.15: pv_0 power in general case when regression coefficients change: $\beta' = \begin{bmatrix} 1 & 3 & 10 & 0 \\ 3 & 5 & 6 & 8 \\ 10 & 0.5 & 5 & 1 \end{bmatrix}$, $\Sigma' = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 5 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 3 \end{bmatrix}$, and $A'' = (1+t)A'$ while Σ and b are fixed.

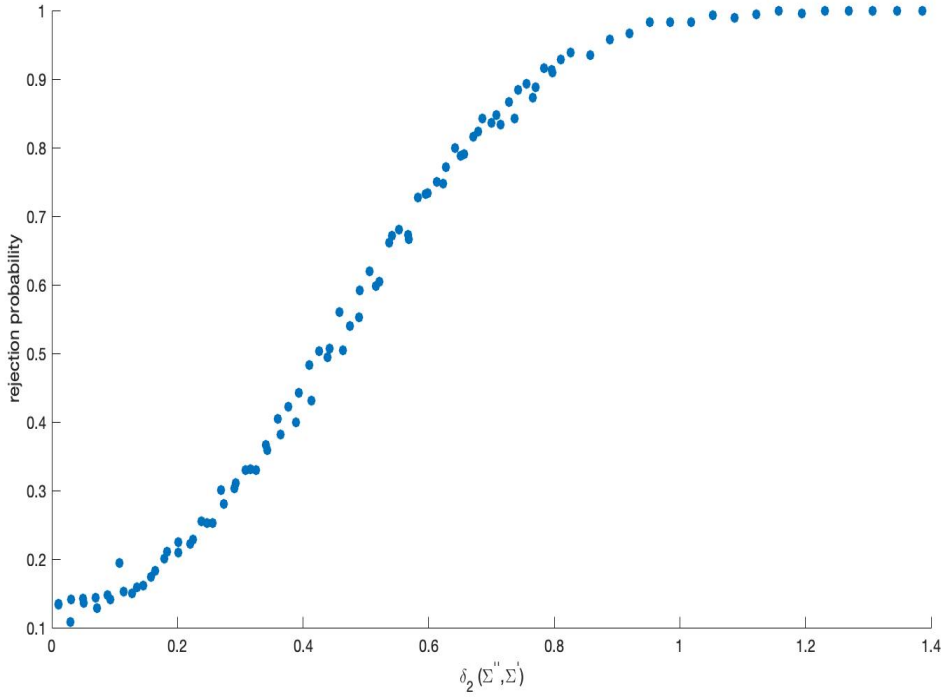


Figure 2.16: pv_0 power in general case when noise changes: we set $\beta' = \begin{bmatrix} 1 & 3 & 10 & 0 \\ 3 & 5 & 6 & 8 \\ 10 & 0.5 & 5 & 1 \end{bmatrix}$, $\Sigma' = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 5 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 3 \end{bmatrix}$, and $\Sigma'' = (1+t)\Sigma'$ while β is fixed.

2.3.2 Real data

2.3.2.1 Setup

In this section, we present a brief overview of the experimental setup. For a comprehensive description, please refer to the methods section in [10].

Two individuals with paralysis (T11 and T5) were enrolled in the BrainGate2 pilot clinical trial. Both had two microelectrode arrays with 96 channels implanted in the area of the precentral gyrus that controls their dominant (left) hand/arm [56]. Neural activity data was extracted from the recordings in windows of 20 milliseconds with no overlap.

As for the neural features, for T5, multi-unit threshold-crossing spike rates ($RMS < -3.5$) per electrode were used. For T11, two types of features are used: both the spike rate (same threshold) and power in the spike-band (250 – 5000 Hz) per electrode. Each

individual neural feature was standardized for each electrode (channel) using a moving window of 3 minutes. This technique, called z-scoring, was the same for both T11 and T5.

T11 completed a closed-loop 2D point-and-click center-out-and-back task over 15 sessions spread across 4 months. T5, on the other hand, closed-loop 2D random target selection task for 6 sessions over 28 days. Due to the nature of the center-out-and-back target acquisition task, true labels (movement intention) (Y) can be inferred at every timestep. These inferred labels, along with the neural recording X are used to find the reported p-value in the next section.

2.3.2.2 Results

The compared data for the result below are each 60-second sliding window updated every 10 seconds. The data of time bins from the first session (T11) or first two sessions (T5) that have angular error $< 4°$ was used to obtain the PCA transformation. Neural features are projected to the top 5 PCs (after z-scoring).

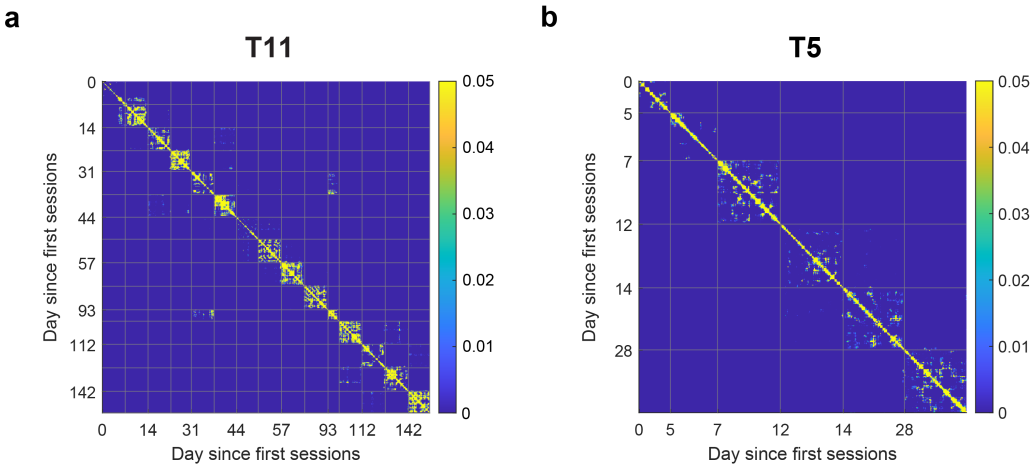


Figure 2.17: Computed p-values (pv_1) between pairs of kinematics-neural data from time bins from sessions, for T11 and T5. Neural data is transformed to the top 5 PC's, extracted from time bins from the first session for T11 that have angular error $< 4°$, and from the first two sessions for T5 that have angular error $< 4°$

Our test results indicate statistically significant shifts in the probabilistic encoding model of neural activity conditioned on the kinematics intention in nearly all compar-

isons between sessions, except for sessions that were completed consecutively (shown diagonally on a graph). This pattern held true for both participants in the study.

2.4 Conclusion

Pun et al. [10] introduced the MINDFUL (Measuring Instabilities in Neural Data for Useful Long-Term iBCI) score to assess changes in decoder performance by measuring the Kullback-Leibler divergence (KLD) between neural feature distributions across time periods. However, this approach may not fully capture changes in the encoding model, especially when movement intention is not fixed. To address this limitation, we proposed two p-values to detect changes in the conditional distribution $p(X|Y)$, which better reflects the underlying encoding process. Importantly, these p-values do not require identical feature distributions ($p(Y)$) between conditions.

The p-value pv_1 demonstrated good power across different paradigms of probabilistic functional mapping changes. However, its limitation lies in its inability to detect differences in covariances between error terms across different dimensions when Σ and Σ' are not diagonal. Specifically, if $A' = A''$ and $\Sigma' jj = \Sigma'' jj$ for all $1 \leq j \leq d$, but there exist $1 \leq j \neq j' \leq d$ such that $\Sigma' jj' \neq \Sigma'' jj'$, pv_1 would not reject the null hypothesis.

Our other proposed p-value, pv_0, is only valid under specific assumptions and does not appear to be robust to deviations from those assumptions. Consequently, pv_0 is not recommended for use unless the diagonal covariance term is believed to be of the form cI.

2.5 Appendix

Proof of Theorem 1:

To start, we first state some well-known lemmas related to the univaruate linear regression model, which we will frequently use. Let $X = y\beta' + Z'$ be the univariate linear Gaussian regression model for a one-dimensional response variable X' given $Y' = y'$, where $y' \in \mathbb{R}^{1\times k}$, $\beta' \in \mathbb{R}^{k\times 1}$, and $Z' \sim N(0, \sigma'^2)$. Let $\mathbf{Y}'$ and $\mathbf{X}'$ respectively be the $n' \times k$ ($k < n'$) design matrix and $n' \times 1$ response matrix. We assume that the corresponding $Z'_1, \ldots, Z'_{n'}$ are independent and unobserved. Let $\mathbf{Z'_{n'}}$ be the $n' \times 1$ random vector $[Z'_1, \ldots, Z'_{n'}]^\top$. Now under the assumption of independence of Z_i's, as well as assumption of homoscedasticity, the univariate model above can be written as

$$\mathbf{X}' = \mathbf{Y}'\beta' + \mathbf{Z}' \tag{2.2}$$

where $\mathbf{Z}' \sim N_{n'}(\mathbf{0}, \sigma'^2 I_{n'})$. Now assuming $\mathbf{Y}'$ has full rank (equivalently, ${\mathbf{Y}'}^\top\mathbf{Y}'$ is invertible), the unbiased estimator for β' is $\widehat{\beta'} = ({\mathbf{Y}'}^\top\mathbf{Y}')^{-1}{\mathbf{Y}'}^\top\mathbf{X}'$.

Lemma 2. *In the univariate model described above and assuming Y has full rank k ($k < n$), the following statements hold:*

(a) The matrix $S = I_{n'} - \mathbf{Y}'({\mathbf{Y}'}^\top\mathbf{Y}')^{-1}{\mathbf{Y}'}^\top$ is symmetric and idempotent with $\mathrm{Tr}(S) = n' - k$.

(b) Let $\widehat{\mathbf{Z}}' = \mathbf{X}' - \mathbf{Y}'\widehat{\beta'}$, then $\widehat{\mathbf{Z}}' = S\mathbf{Z}'$.

(c) For the calculated residual sum of squares $\widehat{\mathbf{R}'}$ ($\widehat{\mathbf{R}'} = {\widehat{\mathbf{Z}}'}^\top\widehat{\mathbf{Z}}'$), we have $\frac{\widehat{\mathbf{R}'}}{\sigma'^2} \sim \chi^2_{n'-k}$.

Proof. (a) It can be easily verified that S is symmetric. We also note that

$$\begin{aligned} S^2 &= (I_{n'} - \mathbf{Y}'({\mathbf{Y}'}^\top\mathbf{Y}')^{-1}{\mathbf{Y}'}^\top)(I_{n'} - \mathbf{Y}'({\mathbf{Y}'}^\top\mathbf{Y}')^{-1}{\mathbf{Y}'}^\top) \\ &= I_{n'} - 2\mathbf{Y}'({\mathbf{Y}'}^\top\mathbf{Y}')^{-1}{\mathbf{Y}'}^\top + \mathbf{Y}'({\mathbf{Y}'}^\top\mathbf{Y}')^{-1}{\mathbf{Y}'}^\top\mathbf{Y}'({\mathbf{Y}'}^\top\mathbf{Y}')^{-1}{\mathbf{Y}'}^\top \\ &= I_{n'} - \mathbf{Y}'({\mathbf{Y}'}^\top\mathbf{Y}')^{-1}{\mathbf{Y}'}^\top \end{aligned}$$

which shows that S is idempotent. As for the trace, we note that

$$\mathrm{Tr}(S) = \mathrm{Tr}(I_{n'}) - \mathrm{Tr}(\mathbf{Y}'(\mathbf{Y}'^{\top}\mathbf{Y}')^{-1}\mathbf{Y}'^{\top}) = n' - \mathrm{Tr}(\mathbf{Y}'^{\top}\mathbf{Y}'(\mathbf{Y}'^{\top}\mathbf{Y}')^{-1}) = n' - \mathrm{Tr}(I_k) = n' - k$$

(b) Note that

$$\widehat{\mathbf{Z}}' = \mathbf{X}' - \mathbf{Y}'\widehat{\beta}' = \mathbf{X}' - \mathbf{Y}'(\mathbf{Y}'^{\top}\mathbf{Y}')^{-1}\mathbf{Y}'^{\top}\mathbf{X}' = S\mathbf{X}' = S(\mathbf{Y}'\beta' + \mathbf{Z}') = S\mathbf{Z}'$$

The last equality holds as $S\mathbf{Y}'\beta' = (I_{n'} - \mathbf{Y}'(\mathbf{Y}'^{\top}\mathbf{Y}')^{-1}\mathbf{Y}'^{\top})\mathbf{Y}'\beta' = \mathbf{Y}'\beta' - \mathbf{Y}'\beta' = 0$

(c) This part is the immediate result of lemma 2.1 in Fisher[74], where $M = S$, while using parts (a) and (b) here. Instead, one can refer to [78] (Chapter 2, Corollary 6) for the proof.

□

Now let us consider another univariate linear Gaussian model as below:

$$\mathbf{X}'' = \mathbf{Y}''\ \beta'' + \mathbf{Z}'' \tag{2.3}$$

for a response variable $\mathbf{X}''$ of size $n'' \times 1$ given design matrix $\mathbf{Y}''$ of size $n'' \times k$, $\mathbf{Z}'' \sim N_{n''}(\mathbf{0}, \sigma''^2 I_{n''})$ and $k < n''$. Now again, assuming $\mathbf{Y}''$ has full rank, the unbiased estimator for β'' is $\widehat{\beta''} = (\mathbf{Y}''^{\top}\mathbf{Y}'')^{-1}\mathbf{Y}''^{\top}\mathbf{X}''$.

Now if we concatenate $\mathbf{X}'$ and $\mathbf{X}''$ in $\mathbf{X}_{comb} = \begin{bmatrix} \mathbf{X}' \\ \mathbf{X}'' \end{bmatrix}$ and we also form the matrix $\mathbf{Y}^* = \begin{bmatrix} \mathbf{Y}' & 0 \\ 0 & \mathbf{Y}'' \end{bmatrix}$, then equations (2.2) and (2.3) can be enclosed as

$$\mathbf{X}_{comb} = \begin{bmatrix} \mathbf{X}' \\ \mathbf{X}'' \end{bmatrix} = \begin{bmatrix} \mathbf{Y}'\beta' \\ \mathbf{Y}''\ \beta'' \end{bmatrix} + \begin{bmatrix} \mathbf{Z}' \\ \mathbf{Z}'' \end{bmatrix} = \mathbf{Y}^* \begin{bmatrix} \beta' \\ \beta'' \end{bmatrix} + \begin{bmatrix} \mathbf{Z}' \\ \mathbf{Z}'' \end{bmatrix} \tag{2.4}$$

Now if $\mathbf{Z}'$ and $\mathbf{Z}''$ are independent, and if $\sigma' = \sigma''$ (which imposes homoscedasticity),

then $\mathbf{Z}_{comb} = \begin{bmatrix} \mathbf{Z}' \\ \mathbf{Z}'' \end{bmatrix}$ is distributed as $N_{n'+n''}(\mathbf{0}, \sigma'^2 I_{n'+n''})$, hence we can use lemma 2 for model in (2.4) (Note that $\mathbf{Y}^*$ is of full rank as well (its rank is $2k$). Now since $2k < n' + n''$). Therefore, $S^* = I_{n'+n''} - \mathbf{Y}^*(\mathbf{Y}^{*\top}\mathbf{Y}^*)^{-1}\mathbf{Y}^{*\top}$ is symmetric and idempotent and $\mathrm{Tr}(S^*) = n' + n'' - 2k$, and we have the following result from lemma 2 part (b):

$$\mathbf{X}_{comb} - \mathbf{Y}^* \begin{bmatrix} \widehat{\beta'} \\ \widehat{\beta''} \end{bmatrix} = S^* \mathbf{Z}_{comb} \tag{2.5}$$

(Note that $\widehat{\begin{bmatrix} \beta' \\ \beta'' \end{bmatrix}} = (\mathbf{Y}^{*\top}\mathbf{Y}^*)^{-1}\mathbf{Y}^{*\top}\mathbf{X}_{comb} = \begin{bmatrix} \widehat{\beta'} \\ \widehat{\beta''} \end{bmatrix}$).

Now Let $\mathbf{Y}_{comb} = \begin{bmatrix} \mathbf{Y}' \\ \mathbf{Y}'' \end{bmatrix}$, and assume that in models (2.2) and (2.3), we have $\beta' = \beta''$ additional to the previous assumptions that $\mathbf{Z}'$ and $\mathbf{Z}''$ are independent and $\sigma' = \sigma''$. Then the model in (2.4) can be written as

$$\mathbf{X}_{comb} = \mathbf{Y}_{comb}\beta' + \mathbf{Z}_{comb} \tag{2.6}$$

and again since $\mathbf{Y}_{comb}$ is of full rank (rank k) and $k < n' + n''$, if $\mathbf{Z}'$ and $\mathbf{Z}''$ are independent and if $\sigma' = \sigma''$ (thereby $\mathbf{Z}_{comb} \sim N_{n'+n''}(\mathbf{0}, \sigma'^2 I_{n'+n''})$), we can use lemma 2 for model in (2.6), and conclude that $S_{comb} = I_{n'+n''} - \mathbf{Y}_{comb}(\mathbf{Y}_{comb}^\top \mathbf{Y}_{comb})^{-1}\mathbf{Y}_{comb}^\top$ is symmetric and idempotent and $\mathrm{Tr}(S_{comb}) = n' + n'' - k$. Moreover, let $\widehat{\beta_{comb}} = (\mathbf{Y}_{comb}^\top \mathbf{Y}_{comb})^{-1}\mathbf{Y}_{comb}^\top \mathbf{X}_{comb}$ be the new estimation of $\beta' = \beta''$ using the combined datasets, then by lemma 2 part (b), we have

$$\mathbf{X}_{comb} - \mathbf{Y}_{comb}\widehat{\beta_{comb}} = S_{comb}\mathbf{Z}_{comb} \tag{2.7}$$

We now have the following lemma:

Lemma 3. *Under the assumptions that was made above, namely, 1)* $\mathbf{Y}$ *and* $\mathbf{Y}''$ *are*

both of full rank, 2) $k < min\{n', n''\}$*, 3)* $\sigma' = \sigma''$*, 4)* $\beta' = \beta''$*, and 5)* $\mathbf{Z}'$ *and* $\mathbf{Z}''$ *are independent, we have the following results:*

(a) Let $\widehat{\mathbf{Z}''} = \mathbf{X}'' - \mathbf{Y}''\,\widehat{\beta''}$ *and* $\widehat{\mathbf{R}''} = \widehat{\mathbf{Z}''}^{\top}\widehat{\mathbf{Z}''}$*, then*

$$\widehat{\mathbf{R}'} + \widehat{\mathbf{R}''} = (S^*\mathbf{Z}_{comb})^{\top}(S^*\mathbf{Z}_{comb})$$

, and $\frac{\widehat{\mathbf{R}'}+\widehat{\mathbf{R}''}}{\sigma'^2}$ *is distributed as* $\chi^2_{n'+n''-2k}$ *(Note that this part does not require assumption 4).*

(b) Let $\widehat{\mathbf{R}_{comb}} = (\mathbf{X}_{comb} - \mathbf{Y}_{comb}\widehat{\beta_{comb}})^{\top}(\mathbf{X}_{comb} - \mathbf{Y}_{comb}\widehat{\beta_{comb}})$*. Then*

$$\widehat{\mathbf{R}_{comb}} - \widehat{\mathbf{R}'} - \widehat{\mathbf{R}''} = \Big((S_{comb} - S^*)\mathbf{Z}_{comb}\Big)^{\top}\Big((S_{comb} - S^*)\mathbf{Z}_{comb}\Big)$$

, and $\frac{\widehat{\mathbf{R}_{comb}}-\widehat{\mathbf{R}}-\widehat{\mathbf{R}''}}{\sigma'^2}$ *is distributed as* χ^2_k*.*

(c) $\frac{\widehat{\mathbf{R}'}+\widehat{\mathbf{R}''}}{\sigma'^2}$ *and* $\frac{\widehat{\mathbf{R}''_{comb}}-\widehat{\mathbf{R}'}-\widehat{\mathbf{R}''}}{\sigma'^2}$ *are independent.*

(d) As a direct result of parts (a), (b) and (c), we have the statistic

$$\frac{n' + n'' - 2k}{k} \cdot \frac{\widehat{\mathbf{R}_{comb}} - \widehat{\mathbf{R}'} - \widehat{\mathbf{R}''}}{\widehat{\mathbf{R}'} + \widehat{\mathbf{R}''}}$$

be distributed as F*-distribution with degrees of freedom* k *and* $n' + n'' - 2k$*.*

Proof. (a) Now note that $\widehat{\mathbf{R}'} + \widehat{\mathbf{R}''} = (\mathbf{X}_{comb} - \mathbf{Y}^* \begin{bmatrix} \widehat{\beta'} \\ \widehat{\beta''} \end{bmatrix})^{\top}(\mathbf{X}_{comb} - \mathbf{Y}^* \begin{bmatrix} \widehat{\beta'} \\ \widehat{\beta''} \end{bmatrix})$, so the result comes directly from lemma 2 parts (b) and (c).

Using equations (2.5) and (2.7), parts (b), (c) and (d) are proved in the proof of lemma 2.2 in Fisher[74], where $M = S_{comb}$, $M^* = S^*$, and it can be easily verified that

$S_{comb}S^* = S^*$, using the fact that $\mathbf{Y}_{comb} = \mathbf{Y}^* A$ where $A = \begin{bmatrix} I_k \\ I_k \end{bmatrix}$:

$$
\begin{aligned}
S_{comb}S^* &= (I_{n'+n''} - \mathbf{Y}_{comb}(\mathbf{Y}_{comb}^\top \mathbf{Y}_{comb})^{-1}\mathbf{Y}_{comb}^\top)(I_{n'+n''} - \mathbf{Y}^*(\mathbf{Y}^{*^\top}\mathbf{Y}^*)^{-1}\mathbf{Y}^{*^\top}) \\
&= (I_{n'+n''} - \mathbf{Y}^* A(A^\top \mathbf{Y}^{*^\top}\mathbf{Y}^* A)^{-1}A^\top\mathbf{Y}^{*^\top})(I_{n'+n''} - \mathbf{Y}^*(\mathbf{Y}^{*^\top}\mathbf{Y}^*)^{-1}\mathbf{Y}^{*^\top}) \\
&= I_{n'+n''} - \mathbf{Y}^*(\mathbf{Y}^{*^\top}\mathbf{Y}^*)^{-1}\mathbf{Y}^{*^\top} \\
&- \mathbf{Y}^* A(A^\top \mathbf{Y}^{*^\top}\mathbf{Y}^* A)^{-1}A^\top\mathbf{Y}^{*^\top} + \mathbf{Y}^* A(A^\top \mathbf{Y}^{*^\top}\mathbf{Y}^* A)^{-1}A^\top\mathbf{Y}^{*^\top}\mathbf{Y}^*(\mathbf{Y}^{*^\top}\mathbf{Y}^*)^{-1}\mathbf{Y}^{*^\top} \\
&= I_{n'+n''} - \mathbf{Y}^*(\mathbf{Y}^{*^\top}\mathbf{Y}^*)^{-1}\mathbf{Y}^{*^\top} \\
&- \mathbf{Y}^* A(A^\top \mathbf{Y}^{*^\top}\mathbf{Y}^* A)^{-1}A^\top\mathbf{Y}^{*^\top} + \mathbf{Y}^* A(A^\top \mathbf{Y}^{*^\top}\mathbf{Y}^* A)^{-1}A^\top\mathbf{Y}^{*^\top} \\
&= I_{n'+n''} - \mathbf{Y}^*(\mathbf{Y}^{*^\top}\mathbf{Y}^*)^{-1}\mathbf{Y}^{*^\top} \\
&= S^*
\end{aligned}
\tag{2.8}
$$

□

Another approach to prove the same results above can be found in Chow[73], p595-598.

We can now turn to the proof of theorem 1 which is for multivariate linear Gaussian model described in section 2.2.

We make frequent use of this basic result.

Lemma 4. *If* $(\mathrm{pv}_i : i \in I)$ *are valid p-values for* H_0 *and* $(\lambda_i : i \in I)$ *are positive constants with* $\sum_i \lambda_i \leq 1$*, then*

$$\mathrm{pv} = \min_{i \in I} \mathrm{pv}_i / \lambda_i$$

is a valid p-value for H_0*.*

Proof. For each $\alpha \in [0,1]$,

$$\begin{aligned}
\mathbb{P}(\text{pv} \leq \alpha) &= \mathbb{P}(\min_{i \in I}(\text{pv}_i/\lambda_i) \leq \alpha) \\
&\leq \sum_{i \in I} \mathbb{P}(\text{pv}_i/\lambda_i \leq \alpha) \\
&= \sum_{i \in I} \mathbb{P}(\text{pv}_i \leq \alpha\lambda_i) \\
&\leq \sum_{i \in I} \alpha\lambda_i \\
&\leq \alpha
\end{aligned}$$

This proves the lemma. □

Note 1: For any valid p-value pv, $\min\{\text{pv}, 1\}$ is also valid. Since for each $\alpha < 1$, $\mathbb{P}(\min\{\text{pv}, 1\} \leq \alpha) = \mathbb{P}(\text{pv} \leq \alpha) \leq \alpha$, and for $\alpha = 1$, $\mathbb{P}(\min\{\text{pv}, 1\} \leq \alpha) \leq 1$.

Note 2:

$$\begin{aligned}
\text{pv}_1 = \min \Big\{ & \lambda_{11}^{-1} F_{n''-k,n'-k}(T_{\Sigma,1}),\ \lambda_{21}^{-1} G_{n''-k,n'-k}(T_{\Sigma,1}),\ \lambda_{31}^{-1} G_{k,n-2k}(T_{\beta,1}), \\
& \lambda_{12}^{-1} F_{n''-k,n'-k}(T_{\Sigma,12}),\ \lambda_{22}^{-1} G_{n''-k,n'-k}(T_{\Sigma,2}),\ \lambda_{32}^{-1} G_{k,n-2k}(T_{\beta,2}), \\
& \dots, \\
& \lambda_{1d}^{-1} F_{n''-k,n'-k}(T_{\Sigma,d}),\ \lambda_{2d}^{-1} G_{n''-k,n'-k}(T_{\Sigma,d}),\ \lambda_{3d}^{-1} G_{k,n-2k}(T_{\beta,d}),\ 1 \Big\}
\end{aligned}$$

These two notes, besides lemma 4 show that to prove that pv_1 is a valid p-value for H_0, it is left to show that for each $j = 1, \dots, d$, $F_{n''-k,n'-k}(T_{\Sigma,j})$, $G_{n''-k,n'-k}(T_{\Sigma,j})$, and $G_{k,n-2k}(T_{\beta,j})$ are valid p-values for H_0. In fact, based on how F_{d_1,d_2} and G_{d_1,d_2} are defined, it suffices to show that under the null hypothesis, $T_{\Sigma,j}$ is distributed as an F distribution with degrees of freedom $n'' - k$ and $n' - k$, and and $T_{\beta,j}$ as F distribution with degrees of freedom k and $n - 2k$ for each j.

Furthermore, by lemma 4 and the first note above, to prove that pv_0 is valid, we are left to show that $T_\Sigma \sim F(d(n'' - k), d(n' - k))$ and $T_\beta \sim F(dk, d(n - 2k))$.

Proof for $T_{\Sigma,j}$:

Note that in the model in equation (2.1) in section 2.2, for each $j \in [1, \ldots, d]$ we have the univariate model of j'th response variable, i.e., the j'th column of X given Y as below:

$$X_{<j>} = y\beta_{<j>} + Z_{<j>} \tag{2.9}$$

where the subscript $< j >$ denotes the j'th column of a matrix or vector, and $Z_{<j>} \sim N_n(\mathbf{0}, \Sigma_{jj} I)$ (In fact, homoscedasticity follows from the observations $Z_1, \ldots, Z_n$ being identically distributed, and non-diagonal 0's arise from the independence of observations).

Therefore, for dataset $(y'_1, X'_1), \ldots, (y'_{n'}, X'_{n'})$ we have $X'_{<j>} = Y'\beta'_{<j>} + Z'_{<j>}$ where $Z'_{<j>} \sim N_{n'}(\mathbf{0}, \Sigma'_{jj} I)$ and for the other dataset, $X''_{<j>} = Y''\beta''_{<j>} + Z''_{<j>}$ with $Z''_{<j>} \sim N_{n''}(\mathbf{0}, \Sigma''_{jj} I)$ for all j.

We also note that for each j, $\widehat{\mathbf{R}}'_{jj}$ defined in section 2.2 is the residual sum of squares associated to the univariate model above, i.e., $\widehat{\mathbf{R}}'_{jj} = (X'_{<j>} - Y'\widehat{\beta_{<j>}'})^\top (X'_{<j>} - Y'\widehat{\beta_{<j>}'})$, as $\widehat{\beta_{<j>}'} = \widehat{\beta'}_{<j>}$. By analogy, $\widehat{\mathbf{R}}''_{jj} = (X''_{<j>} - Y''\widehat{\beta_{<j>}''})^\top (X''_{<j>} - Y''\widehat{\beta_{<j>}''})$. Now since by the assumptions of section 2.2, Y' and Y'' both have full rank and $k < min\{n', n''\}$, we can use lemma 2 part (c) to conclude that $\frac{\widehat{\mathbf{R}}'_{jj}}{\Sigma_{jj}'} \sim \chi^2_{n'-k}$ and $\frac{\widehat{\mathbf{R}}''_{jj}}{\Sigma_{jj}''} \sim \chi^2_{n''-k}$. Also as a result of the other assumption in section 2.2 that two datasets are conditionally independent given Y' and Y'' (i.e., $p(X', X''|Y', Y'') = p(X'|Y')p(X''|Y'')$, thus $p(Z', Z'') = p(Z')p(Z'')$), therefore Z' and Z'' are independent, then $\widehat{\mathbf{R}}'_{jj}$ and $\widehat{\mathbf{R}}''_{jj}$ are independent by lemma 2 parts (b) and (c).

Now under H_0, we have $\Sigma_{jj}' = \Sigma_{jj}''$ for all j. Therefore, the desired result that $T_{\Sigma,j} \sim F_{n''-k,n'-k}$ under H_0 is derived.

Proof for $T_{\beta,j}$:

As for $T_{\beta,j}$, we note that $\widehat{\mathbf{R}}_{jj}$ defined in section 2.2 is equal to $(X_{<j>}-Y\widehat{\beta}_{<j>})^\top(X_{<j>}-Y\widehat{\beta}_{<j>})$, where $X(Y)$ was the combined data of X' and X'' (Y' and Y''), and $\hat{\beta}$ was estimated from the datasets combined. In fact, this is coming from the combined univariate model in (2.6), and the j'th column of $\widehat{\beta}$ is $\widehat{\beta}_{comb}$ in (2.7). Under H_0 since $\beta'_{<j>} = \beta''_{<j>}$ for all j, and as all other conditions of lemma 3 hold, we can use parts (b) and then (d), to obtain the desired result that $T_{\beta,j} \sim F_{k,n'+n''-2k}$ under H_0 for all $j = 1, \ldots, d$.

This completes the proof that pv_1 is a valid p-value.

Remark 1. : Note that H_0 is a strict subset of the set of model class distributions where $\beta' = \beta''$ and ${\Sigma_{jj}}' = {\Sigma_{jj}}''$ for all j, and this is why this test does not have good power on detection of cases where the mentioned conditions hold, but $\Sigma'_{ij} \neq \Sigma''_{ij}$ for some $i \neq j$.

:

Now to prove the theorem for pv_0 we need the lemma below.

Lemma 5. *: If Σ is diagonal in the model* (2.1)*, then $\widehat{\mathbf{R}}_{ii}$ and $\widehat{\mathbf{R}}_{jj}$ are independent for all $i \neq j \in [1, \ldots, d]$.*

Proof. As $\widehat{\mathbf{R}}_{jj} = (X_{<j>} - Y\widehat{\beta}_{<j>})^\top(X_{<j>} - Y\widehat{\beta}_{<j>})$, by lemma 2 part (b), to prove independence of $\widehat{\mathbf{R}}_{ii}$ and $\widehat{\mathbf{R}}_{jj}$, it suffices to show independence of $Z_{<i>}$ and $Z_{<j>}$, which is true as they are uncorrelated (when Σ is diagonal) and multivariate Gaussian ($Z_{<i>} \sim N_n(\mathbf{0}, \Sigma_{ii}I)$). □

Proof for T_Σ:

If $\beta' = \beta''$, and we additionally have $\Sigma' = \Sigma'' = cI$, then using the results we obtained in the part for $T_{\Sigma,j}$ above, we have $\frac{\widehat{\mathbf{R}}'_{jj}}{c} \sim \chi^2_{n'-k}$ and $\frac{\widehat{\mathbf{R}}''_{jj}}{c} \sim \chi^2_{n''-k}$ for all j. Now by lemma 5, $\widehat{\mathbf{R}}'_{ii}$ and $\widehat{\mathbf{R}}'_{jj}$ are independent for $i \neq j \in [1, \ldots, d]$, therefore, $\sum_{j=1}^d \frac{\widehat{\mathbf{R}}'_{jj}}{c} \sim \chi^2_{d(n'-k)}$, and analogously we have $\sum_{j=1}^d \frac{\widehat{\mathbf{R}}''_{jj}}{c} \sim \chi^2_{d(n''-k)}$. As the two

datasets are independent, so are $\widehat{\mathbf{R}}'$ and $\widehat{\mathbf{R}}''$, we have the desired result that

$$T_\Sigma = \frac{\sum_{j=1}^{d} \frac{\widehat{\mathbf{R}}''_{jj}}{cd(n''-k)}}{\sum_{j=1}^{d} \frac{\widehat{\mathbf{R}}'_{jj}}{cd(n'-k)}}$$

is distributed as $F(d(n''-k), d(n'-k))$.

Proof for T_β:

Under the assumption that $\beta' = \beta''$, and if we additionally have $\Sigma' = \Sigma''$, we can write $X = Y\beta' + Z$ where $Z \sim N_d(\mathbf{0}, \Sigma')$ for combined data $Y = \begin{bmatrix} Y' \\ Y'' \end{bmatrix}$ and $X = \begin{bmatrix} X' \\ X'' \end{bmatrix}$. Then if $\Sigma' = \Sigma'' = cI$, using lemma 3, parts (a) and (b) we have these results that $\frac{\widehat{\mathbf{R}}'_{jj}+\widehat{\mathbf{R}}''_{jj}}{c} \sim \chi^2_{n-2k}$ and $\frac{\widehat{\mathbf{R}}_{jj}-\widehat{\mathbf{R}}'_{jj}-\widehat{\mathbf{R}}''_{jj}}{c} \sim \chi^2_k$. Plus, from the same parts of that lemma we also have

$$\widehat{\mathbf{R}}'_{jj} + \widehat{\mathbf{R}}''_{jj} = (S^* Z_{<j>})^\top (S^* Z_{<j>})$$

where $S^* = I_n - Y^*({Y^*}^\top Y^*)^\top {Y^*}^\top$, and $Y^* = \begin{bmatrix} Y' & 0 \\ 0 & Y'' \end{bmatrix}$, and

$$\widehat{\mathbf{R}}_{jj} - \widehat{\mathbf{R}}'_{jj} + \widehat{\mathbf{R}}''_{jj} = ((S_{comb} - S^*)Z_{<j>})^\top ((S_{comb} - S^*)Z_{<j>})$$

where $S_{comb} = I_n - Y(Y^\mathrm{T}Y)^\top Y^\mathrm{T}$. Therefore, by the same argument as in the proof of lemma 5, when $\Sigma' = \Sigma''$ (hence $Z \sim N_d(\mathbf{0}, \Sigma')$) and it is diagonal, we have independence of $Z_{<i>}$ and $Z_{<j>}$ for each $i \neq j \in [1, \ldots, d]$, which is followed by independence of $\widehat{\mathbf{R}}'_{ii}+\widehat{\mathbf{R}}''_{ii}$ and $\widehat{\mathbf{R}}'_{jj}+\widehat{\mathbf{R}}''_{jj}$, as well as independence of $\widehat{\mathbf{R}}_{ii}-\widehat{\mathbf{R}}'_{ii}+\widehat{\mathbf{R}}''_{ii}$ and $\widehat{\mathbf{R}}_{jj}-\widehat{\mathbf{R}}'_{jj}+\widehat{\mathbf{R}}''_{jj}$. Hence we have $\sum_{j=1}^{d} \frac{\widehat{\mathbf{R}}_{jj}-\widehat{\mathbf{R}}'_{jj}-\widehat{\mathbf{R}}''_{jj}}{c} \sim \chi^2_{dk}$, and $\sum_{j=1}^{d} \frac{\widehat{\mathbf{R}}'_{jj}+\widehat{\mathbf{R}}''_{jj}}{c} \sim \chi^2_{d(n-2k)}$.

Now, if we show that these two sums are independent, then

$$T_\beta = \frac{\sum_{j=1}^{d} \frac{\widehat{\mathbf{R}}_{jj} - \widehat{\mathbf{R}}'_{jj} - \widehat{\mathbf{R}}''_{jj}}{cdk}}{\sum_{j=1}^{d} \frac{\widehat{\mathbf{R}}'_{jj} + \widehat{\mathbf{R}}''_{jj}}{cd(n-2k)}}$$

is distributed as $F(dk, d(n-2k))$.

Therefore, to complete the proof, we only need to prove the following lemma:

Lemma 6. *Under H_0 and additional assumption that $\Sigma' = \Sigma'' = cI$, $\sum_{j=1}^{d}(\widehat{\mathbf{R}}_{jj} - \widehat{\mathbf{R}}'_{jj} - \widehat{\mathbf{R}}''_{jj})$ and $\sum_{j=1}^{d}(\widehat{\mathbf{R}}'_{jj} + \widehat{\mathbf{R}}''_{jj})$ are independent.*

Proof. Let $\mathcal{Z} = [Z_{<1>}^\top Z_{<2>}^\top \cdots Z_{<d>}^\top]^\top$ be the concatenated random vector of columns of Z. Under the assumption that $\Sigma' = \Sigma'' = cI$, we have $\mathcal{Z} \sim N_{dn}(\mathbf{0}, cI_{dn})$. Let $\mathcal{S}^*$ and $\mathcal{S}_{comb}$ be defined this way:

$$\mathcal{S}^* = \left[\begin{array}{c|c|c|c} S^* & 0 & \cdots & 0 \\ \hline 0 & S^* & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & S^* \end{array}\right]_{dn \times dn}$$

$$\mathcal{S}_{comb} = \left[\begin{array}{c|c|c|c} S_{comb} & 0 & \cdots & 0 \\ \hline 0 & S_{comb} & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & S_{comb} \end{array}\right]_{dn \times dn}$$

then $\sum_{j=1}^{d}(\widehat{\mathbf{R}}'_{jj} + \widehat{\mathbf{R}}''_{jj}) = (\mathcal{S}^*\mathcal{Z})^\top(\mathcal{S}^*\mathcal{Z})$ and $\sum_{j=1}^{d}(\widehat{\mathbf{R}}_{jj} - \widehat{\mathbf{R}}'_{jj} - \widehat{\mathbf{R}}''_{jj}) = \Big((\mathcal{S}_{comb} - \mathcal{S}^*)\mathcal{Z}\Big)^\top \Big((\mathcal{S}_{comb} - \mathcal{S}^*)\mathcal{Z}\Big)$. Therefore, to show the desired result, it suffices to show that $\mathcal{S}^*\mathcal{Z}$ and $(\mathcal{S}_{comb} - \mathcal{S}^*)\mathcal{Z}$ are independent. Note that both of these random vectors have mean zero, so we have their covariance matrix below:

$$
\begin{aligned}
cov\Big(\mathcal{S}^*\mathcal{Z}, (\mathcal{S}_{comb} - \mathcal{S}^*)\mathcal{Z}\Big) &= \mathbb{E}\left[\mathcal{S}^*\mathcal{Z}\Big((\mathcal{S}_{comb} - \mathcal{S}^*)\mathcal{Z}\Big)^\top\right] \\
&= \mathbb{E}\left[\mathcal{S}^*\mathcal{Z}\mathcal{Z}^\top(\mathcal{S}_{comb} - \mathcal{S}^*)^\top\right] \\
&= \mathcal{S}^*\,\mathbb{E}\left[\mathcal{Z}\mathcal{Z}^\top\right](\mathcal{S}_{comb} - \mathcal{S}^*)^\top \\
&= c\mathcal{S}^*(\mathcal{S}_{comb} - \mathcal{S}^*)^\top \\
&= \mathcal{S}^*\mathcal{S}_{comb} - \mathcal{S}^{*^2} \\
&= 0
\end{aligned}
$$

where the one before the last equality comes from S^* and S_{comb} being symmetric and idempotent as showed earlier, and from equation (2.8), all of which can be easily verified that are extendable to the blocked matrices $\mathcal{S}^*$ and $\mathcal{S}_{comb}$.

Now since $\mathcal{S}^*\mathcal{Z}$ and $(\mathcal{S}_{comb} - \mathcal{S}^*)\mathcal{Z}$ are both multivariate Gaussian, we are done showing their independence.

□

CHAPTER 3

Model drift and its relationship to decoding performance in brain-computer interfaces

3.1 Introduction

Multiple studies have observed a persistent decline in BCI decoder performance after initial calibration sessions [10, 15–20]. While the encoding model's assumed stochastic noise can introduce instantaneous decoding errors, such noise alone is insufficient to explain the observed persistent reduction in average decoding performance over time (lasting seconds or more). This observed instability likely arises from statistical differences between the joint distributions of neural signals and motor intentions at training and test times. Otherwise, the law of large numbers would ensure that any trained decoder with sufficient training data should maintain its average training performance for extended test time.

The previous chapter explored how the conditional distribution of neural ensemble activity for a given motor intention (encoding model) can change over time. We developed a statistical methodology to detect these changes, and real-world data from

two individuals with paralysis confirmed statistically significant alterations between recordings from non-consecutive sessions.

A straightforward solution to address these model drifts might be to retrain the decoder every time a change is detected by our multiple hypothesis test. However, this approach is impractical in clinical settings for two key reasons. First, retraining frequently disrupts user experience and can be burdensome. Second, our test, like many existing methods [24, 75, 76], requires access to both the user's neural recording and their intended movements, which may not be readily available during free BCI use, where the user controls a cursor or other interface without explicit cues for movements. Consequently, the "true labe" (intended movement) cannot be reliably inferred.

However, the primary concern in BCI systems is maintaining good performance of the initially trained decoder over time, even if the underlying encoding model changes. If we can estimate performance based on a readily computable measure of model drift, we can automatically trigger retraining before prediction errors become unacceptable. Pun et al. [10] explored this concept with their proposed MINDFUL (Measuring Instabilities in Neural Data for Useful Long-Term iBCI) score, which serves as a surrogate to measure the changes in performance. The MINDFUL score calculates a statistical distance, which is Kullback-Leibler divergence (KLD), between the marginal distribution of recorded neural features in two different time periods. One period is the target epoch where we want to predict performance, and the other is a reference epoch where the decoder's performance is known to be good. Pun et al. [10] showed a significant correlation between this KLD, which measures the difference between the neural activity patterns in these epochs, and the actual performance of the initially trained decoder at the target epoch, measured by median angular error.

In fact, their experimental results suggest that the performance degradation caused by model drift can be almost entirely explained by the difference in the marginal distributions of neural activity. This real-world observation motivates the central question addressed in this chapter: For a linear Gaussian encoding model, how does the Guassian Kullback-

Leibler divergence (GKLD) between neural data from two different epochs vary with changes in the distribution of prediction error components? We specifically address this question under the assumption of the mathematically optimal decoding scheme.

The first upcoming section investigates the relationship between the GKLD of neural data and the GKLD of the prediction error from the optimal decoder. We explore this connection under various perturbation paradigms applied to the linear-Gaussian encoding model, as outlined in [17]. Furthermore, we utilize simulations to explore the impact of these perturbations on specific components of the prediction error, such as the angular error or the l_2 norm of the error.

The second section focuses on a scenarios where no underlying encoding model drift occurs. In such cases, the sole contributor to decoding errors is instantaneous noise. We investigate how GKLD between the neural data given a particular angular error and the neural data given small error changes with the magnitude of the angular error itself.

Building upon the previous sections, the third section explores the contributions of both noise and model drift on GKLD. We analyze how GKLD behaves under the influence of noise alone, model drift alone, and both factors acting simultaneously. This comprehensive analysis provides a deeper understanding of how these factors interact and ultimately affect decoding performance.

3.2 How neural distribution changes with error

3.2.1 The encoding model

Let $\mathbb{E}[X|Y] = AY + b$ be the linear regression encoding model of neural data given kinematics, where X is the matrix of neural data of size $d \times 1$, Y is movement intentions of size $l \times 1$, A is the regression coefficients matrix of size $d \times l$, and we assume a multidimensional Gaussian noise Z for this model, i.e.,

$$X = AY + b + Z \tag{3.1}$$

where Z is assumed to be independent of Y, and is distributed as $N_d(\mathbf{0}, \Sigma)$.

If Y is distributed as $N_l(\mu, \Gamma)$, the distribution of X will be known, i.e., as $Z = Uz$ for some matrix U and a random vector z of iid standard normal random variables, any linear combination of coordinates of random vector X will be normal, so it is multivariate Gaussian, with

$$\mathbb{E}[X] = A\,\mathbb{E}[Y] + b + \mathbb{E}[Z] = A\mu + b \tag{3.2}$$

And covariance matrix:

$$\begin{aligned}
\text{cov}(X) &= \mathbb{E}[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^\top] \\
&= \mathbb{E}[(X - \mathbb{E}[X])(X^\top - \mathbb{E}[X]^\top)] \\
&= \mathbb{E}[(A(Y - \mu) + Z)((Y^\top - \mu^\top)A^\top + Z^\top)] \\
&= \mathbb{E}[A(Y - \mu)(Y^\top - \mu^\top)A^\top] + \mathbb{E}[A(Y - \mu)Z^\top] + \mathbb{E}[Z(Y^\top - \mu^\top)A^\top] + \mathbb{E}[ZZ^\top] \\
&\overset{Y \perp Z}{=} A\,\mathbb{E}[(Y - \mu)(Y^\top - \mu^\top)]A^\top + 0 + 0 + \Sigma \\
&= A\Gamma A^\top + \Sigma
\end{aligned} \tag{3.3}$$

Therefore, $X \sim N_d(A\mu + b, A\Gamma A^\top + \Sigma)$

3.2.2 The optimal decoder

Lemma 7. *Let $X = AY + b + Z$, where Z has mean zero and $\text{cov}(Z) = \Sigma$, and $\mathbb{E}[Y] = \mu$ and $\text{cov}(Y) = \Gamma$. Then the optimal linear decoder $g(X)$ minimizing the loss function $\mathbb{E}[||Y - g(X)||^2]$ is*

$$\widehat{Y} = g(X) = PX + (I - PA)\mu - Pb \tag{3.4}$$

for $P = \Gamma A^\top (A\Gamma A^\top + \Sigma)^{-1}$.

Proof can be found in the Appendix.

Corollary 8. *In case that* $\mathbb{E}[Y] = \mathbf{0}$*,*

$$\widehat{Y} = g(X) = \Gamma A^\top (A\Gamma A^\top + \Sigma)^{-1}(X - b) \tag{3.5}$$

And if additionally, $cov(Y) = I$*, then the mapping* g *will only depend on the parameters on conditional distribution of* X *given* Y*, namely,* A*,* b*, and* Σ*.*

$$\widehat{Y} = g_{A,b,\Sigma}(X) = P_{A,\Sigma}(X - b) \tag{3.6}$$

where $P_{A,\Sigma} = A^\top (AA^\top + \Sigma)^{-1}$.

3.2.3 Analysis of change in error for using a wrong decoder

In the previous section, we showed that we can predict Y, represented by $\widehat{Y}$, as $g(X)$, where $g(X)$ is the optimal decoder minimizing the mean squared prediction error. For all analysis throughout this section, we assume that $Y \sim N(\mathbf{0}, I)$. Therefore, by corollary 8 we know that under this assumption, the map $g_{A,b,\Sigma}(X) = P_{A,\Sigma}(X - b)$ denotes the optimal decoder for the model described in (3.1), where $P_{A,\Sigma} = A^\top (AA^\top + \Sigma)^{-1}$.

Now we assume that the encoder has perturbed from its original version, and the new encoder is modeled by another linear Gaussian parameterized by $\overline{A}$, $\overline{b}$, and $\overline{\Sigma}$. Let $\overline{X}$ denote the neural data modeled by this new encoder, then $\mathbb{E}[\overline{X}|Y] = \overline{A}Y + \overline{b}$, and we have the Gaussian noise $\overline{Z} \sim N_d(\mathbf{0}, \overline{\Sigma})$ as below:

$$\overline{X} = \overline{A}Y + \overline{b} + \overline{Z} \tag{3.7}$$

Now if we decode $\overline{X}$ using the decoder $g_{A,b,\Sigma}$ designed based on the original model in (3.1), then the estimated movement intention will be $\widehat{\overline{Y}} = g_{A,b,\Sigma}(\overline{X}) = P_{A,\Sigma}(\overline{X} - b) = P_{A,\Sigma}(\overline{A}Y + \overline{b} - b) + P_{A,\Sigma}\overline{Z}$.

It can also be easily verified that given Y, $\widehat{Y}$ and $\widehat{\overline{Y}}$ are distributed as $N(P_{A,\Sigma}AY, P_{A,\Sigma}\Sigma P_{A,\Sigma}^\top)$ and $N(P_{A,\Sigma}(\overline{A}Y + \overline{b} - b), P_{A,\Sigma}\overline{\Sigma}P_{A,\Sigma}^\top)$ respectively.

3.2.3.1 Distribution of prediction error

The prediction error while using the correct optimal decoder is $\widehat{Y} - Y$, and is $\widehat{\overline{Y}} - Y$ if the underlying model has changed, but the neural data is still being decoded by the same decoder $g_{A,\Sigma}$. Hence,

$$\widehat{\overline{Y}} - Y = P_{A,\Sigma}(\overline{A}Y + \overline{b} - b) + P_{A,\Sigma}\overline{Z} - Y = (P_{A,\Sigma}\overline{A} - I)Y + P_{A,\Sigma}(\overline{b} - b) + P_{A,\Sigma}\overline{Z} \quad (3.8)$$

Again since 1) $Y \sim N(\mathbf{0}, I)$ and 2) $\overline{Z} = \overline{U}\overline{z}$ for some matrix $\overline{U}$ and random vector $\overline{z}$ with iid standard normal random variable coordinates, it follows that $\widehat{\overline{Y}} - Y$ is multivariate Gaussian with mean vector and covariance matrix as below:

$$\mathbb{E}[\widehat{\overline{Y}} - Y] = P_{A,\Sigma}(\overline{b} - b) \quad (3.9)$$

and

$$\text{cov}(\widehat{\overline{Y}} - Y) = (P_{A,\Sigma}\overline{A} - I)(P_{A,\Sigma}\overline{A} - I)^\top + P_{A,\Sigma}\overline{\Sigma}{P_{A,\Sigma}}^\top \quad (3.10)$$

And similarly:

$$\mathbb{E}[\widehat{Y} - Y] = 0 \quad (3.11)$$

and

$$\text{cov}(\widehat{Y} - Y) = (P_{A,\Sigma}A - I)(P_{A,\Sigma}A - I)^\top + P_{A,\Sigma}\Sigma{P_{A,\Sigma}}^\top \quad (3.12)$$

3.2.3.2 Change in Gaussian KL distances of errors with certain changes of encoding model

3.2.3.2.1 Gaussian KL distance analysis As we discussed in the previous section 3.2.3.1, $\widehat{\overline{Y}} - Y$ and $\widehat{Y} - Y$ are both multivariate Gaussian, so we can find the Kullback–Leibler divergence between two multivariate Gaussians of $N_k(\mu_0, \Sigma_0)$ and $N_k(\mu_1, \Sigma_1)$ using the formula below:

$$d_{GKL}\big(N(\mu_0, \Sigma_0)||N(\mu_1, \Sigma_1)\big) = \frac{1}{2}\left(\text{Tr}(\Sigma_1^{-1}\Sigma_0) - k + (\mu_1 - \mu_0)^\top \Sigma_1^{-1}(\mu_1 - \mu_0) + \ln\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right) \right) \quad (3.13)$$

We are interested to investigate how $d_{GKL}(\overline{X}||X)$ changes with $d_{GKL}(\widehat{\overline{Y}}-Y||\widehat{Y}-Y)$ for different scenarios of perturbation of the encoding model, as suggested in [17] and [15], listed as follows: base change, noise change, rotation of preferred direction, change of amplitude, and dropping channels.

We are also interested in analysis of error in terms of angular error and the error norm. Note that the distribution of angular error are not necessarily Gaussian, however, we still use the formula (3.13) to estimate the Gaussian KL divergence.

To address how much the Gaussian KL distance (GKLD) of distributions of error norm changes with Gaussian KL distance of neural data, we simulate errors $\widehat{Y}-Y$ and, for each perturbation of the encoding model, the associated $\widehat{\overline{Y}}-Y$ as both their distribution are known (section 3.2.3.1), then we compute the norms and then assuming Gaussian, we find the Gaussian KL distance between the distribution of norms. Instead of this Monte-Carlo approach, since the distribution of $||\widehat{Y}-Y||^2$ is known by the following lemma (see [78] for proof), we can use the analytical mean and variance as well.

Lemma 9. *Let Y is distributed as multivariate Gaussian with mean μ and covariance matrix Σ. Let $\Sigma = P^{\mathrm{T}}\Delta P$ be the orthogonal diagonalization of matrix Σ, $U = P\Sigma^{-1/2}(Y-\mu)$, and $b = P\Sigma^{-1/2}\mu$, then U is distributed as multivariate Gaussian with mean $\mathbf{0}$ and covariance matrix I, and $||Y||^2 = \sum_{i=1}^{n}\lambda_i(U_i+b_i)^2$, where $\{\lambda_1,\cdots,\lambda_n\}$ is the spectrum of matrix Σ (the diagonal of Δ). Namely, $||Y||^2$ is a linear combination of independent non-central chi-squared variables, each with one degree of freedom.*

As for analysis of the angular error, we first simulate Y, make the evoked neural data X by this original encoding model given Y , and then we find $\widehat{Y}$ as a function of X, using the optimal decoder found in section 3.2.2. Then for each perturbation, we make the new evoked neural data $\overline{X}$ by this new encoding model given Y, followed by computing $\widehat{\overline{Y}}$ as a same function of $\overline{X}$ throug using the same decoder, and then we compute the angles between Y and its both predictions ($\widehat{Y}$ and $\widehat{\overline{Y}}$).

To start with, it is worth writing down the Gaussian KL distance between the distribution of prediction errors $\widehat{Y} - Y$ and $\widehat{\overline{Y}} - Y$, i.e., $d_{GKL}(\widehat{\overline{Y}} - Y||\widehat{Y} - Y)$ which is equal to:

$$\begin{aligned}
&\frac{1}{2}\Bigg\{ \mathrm{Tr}\left(\Big((PA-I)(PA-I)^\top + P\Sigma P^\top\Big)^{-1}\Big(\big(P\overline{A}-I\big)\big(P\overline{A}-I\big)^\top + P\overline{\Sigma}P^\top\Big)\right) \\
&- l + \left(\Big(-P(\overline{b}-b)\Big)^\top\Big((PA-I)(PA-I)^\top + P\Sigma P^\top\Big)^{-1}\Big(-P\left(\overline{b}-b\right)\Big)\right) \\
&+ \ln\left(\frac{\det\Big((PA-I)(PA-I)^\top + P\Sigma P^\top\Big)}{\det\Big(\big(P\overline{A}-I\big)\big(P\overline{A}-I\big)^\top + P\overline{\Sigma}P^\top\Big)}\right)\Bigg\}
\end{aligned} \tag{3.14}$$

where $P = P_{A,\Sigma}$.

And similarly, the Gaussian KL distance between neural data of the original encoder X and the perturbed encoder $\overline{X}$, i.e., $d_{GKL}(\overline{X}||X)$ is given by:

$$\begin{aligned}
d_{GKL}(\overline{X}||X) = &\frac{1}{2}\left(\mathrm{Tr}\left((AA^\top+\Sigma)^{-1}(\overline{A}\,\overline{A}^\top+\overline{\Sigma})\right) + (b-\overline{b})^\top(AA^\top+\Sigma)^{-1}(b-\overline{b})\right) \\
&+ \frac{1}{2}\left(\ln\left(\frac{\det(AA^\top+\Sigma)}{\det(\overline{A}\,\overline{A}^\top+\overline{\Sigma})}\right)\right) \\
&- \frac{1}{2}(d)
\end{aligned} \tag{3.15}$$

From here on, we fix $l = 3$ and $d = 20$. For analysis of norm and angular error where simulation is needed, the number used for Monte-Carlo simulation is 10^5. Also, the parameters of the original encoder (A, b, and Σ) are chosen randomly in the beginning, and then for each perturbation paradigm, we are changing one (or two in the case of dropping channels) of parameters A, b, or Σ, while keeping the other parameters fixed.

1. **Base Change** Any linear perturbation of the base firing rates can be written as $\overline{b} = tb + (1-t)b^*$ for t changing from 1 to 0, where b is the d dimensional baseline of the original model and b^* is any vector of size d.

   Lemma 10. *With the linear perturbation of the baseline as above,* $d_{GKL}(\widehat{\overline{Y}} - Y||\widehat{Y} - Y)$ *changes with* $d_{GKL}(\overline{X}||X)$ *linearly.*

Proof. Let $\overline{b}_s = t_s b + (1-t_s)b^*$, $\overline{X}_s$ be the neural data for an encoder model with parameters $A, \overline{b}_s, \Sigma$, and $\widehat{\overline{Y}}_s$ be the estimated kinematics as a function $g_{A,b,\Sigma}$ of $\overline{X}_s$, where A, b, Σ are parameters of the original model. For $i \neq j \in [0,1]$ we have:

$$\begin{aligned}
&\frac{d_{GKL}(\widehat{\overline{Y}}_j - Y||\widehat{Y} - Y) - d_{GKL}(\widehat{\overline{Y}}_i - Y||\widehat{Y} - Y)}{d_{GKL}(\overline{X}_j||X) - d_{GKL}(\overline{X}_i||X)} \\
&= \frac{(-P_{A,\Sigma}(\overline{b}_j - b))^\top M_{A,\Sigma}(-P_{A,\Sigma}(\overline{b}_j - b))}{(b-\overline{b}_j)^\top N_{A,\Sigma}(b-\overline{b}_j) - (b-\overline{b}_i)^\top N_{A,\Sigma}(b-\overline{b}_i)} \\
&- \frac{(-P_{A,\Sigma}(\overline{b}_i - b))^\top M_{A,\Sigma}(-P_{A,\Sigma}(\overline{b}_i - b))}{(b-\overline{b}_j)^\top N_{A,\Sigma}(b-\overline{b}_j) - (b-\overline{b}_i)^\top N_{A,\Sigma}(b-\overline{b}_i)} \\
&= \frac{(1-t_j)^2(P_{A,\Sigma}(b-b^*))^\top M_{A,\Sigma}(P_{A,\Sigma}(b-b^*))}{(1-t_j)^2(b-b^*)^\top N_{A,\Sigma}(b-b^*) - (1-t_i)^2(b-b^*)^\top N_{A,\Sigma}(b-b^*)} \\
&- \frac{(1-t_i)^2(P_{A,\Sigma}(b-b^*))^\top M_{A,\Sigma}(P_{A,\Sigma}(b-b^*))}{(1-t_j)^2(b-b^*)^\top N_{A,\Sigma}(b-b^*) - (1-t_i)^2(b-b^*)^\top N_{A,\Sigma}(b-b^*)} \\
&= \frac{(P_{A,\Sigma}(b-b^*))^\top M_{A,\Sigma}(P_{A,\Sigma}(b-b^*))}{(b-b^*)^\top N_{A,\Sigma}(b-b^*)}
\end{aligned} \tag{3.16}$$

where

$$M_{A,\Sigma} = ((P_{A,\Sigma}A - I)(P_{A,\Sigma}A - I)^\top + P_{A,\Sigma}\Sigma {P_{A,\Sigma}}^\top)^{-1}$$

and

$$N_{A,\Sigma} = (AA^\top + \Sigma)^{-1}$$

both depend on A and Σ, which results in linear relation between $d_{GKL}(\widehat{\overline{Y}} - Y||\widehat{Y} - Y)$ and $d_{GKL}(\overline{X}||X)$ for a fixed original model parameterized by A, b, Σ. □

This phenomena is also verified in the figure below, where $\overline{b}$ is changing from b to some random b^*. It is worth mentioning that no simulation is done for this figure, as the distributions of neural data and error vectors is known (hence we can directly use equations (3.14) and (3.15)). In the figures below, each point corresponds to a certain $t_s \in [0,1]$, with x coordinate of $d_{GKL}(\overline{X}_s||X)$ and y coordinate of $d_{GKL}(\widehat{\overline{Y}}_s - Y||\overline{Y}_s - Y)$. t_s takes 90 equally distanced values in $[0,1]$. Also, b^* is chosen randomly. Computing the slope that we found in (3.16), we obtain 1.8541, which is precisely the slope of figure below.

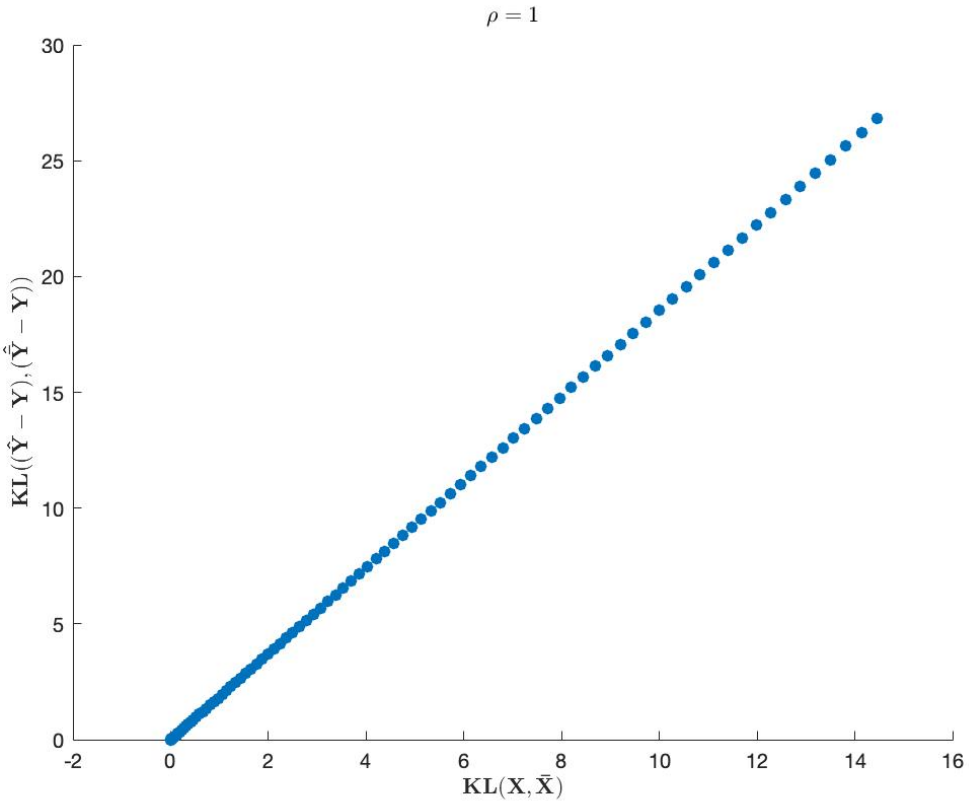


Figure 3.1: How distribution of neural data changes with distribution of error, when using the same decoder and while base changes. No Simulation.

As we discussed earlier, simulation is used to gain the Gaussian KL distance between the angle and error norm. We represent the results below, where the plots from left, respectively show the change of first, the Gaussian KL distance between distributions of angular error, second, the change in the mean of angular error, and third, the Gaussian KL distance of the error norm change with the Gaussian KL distance of neural data, for the same perturbation of baseline as above.

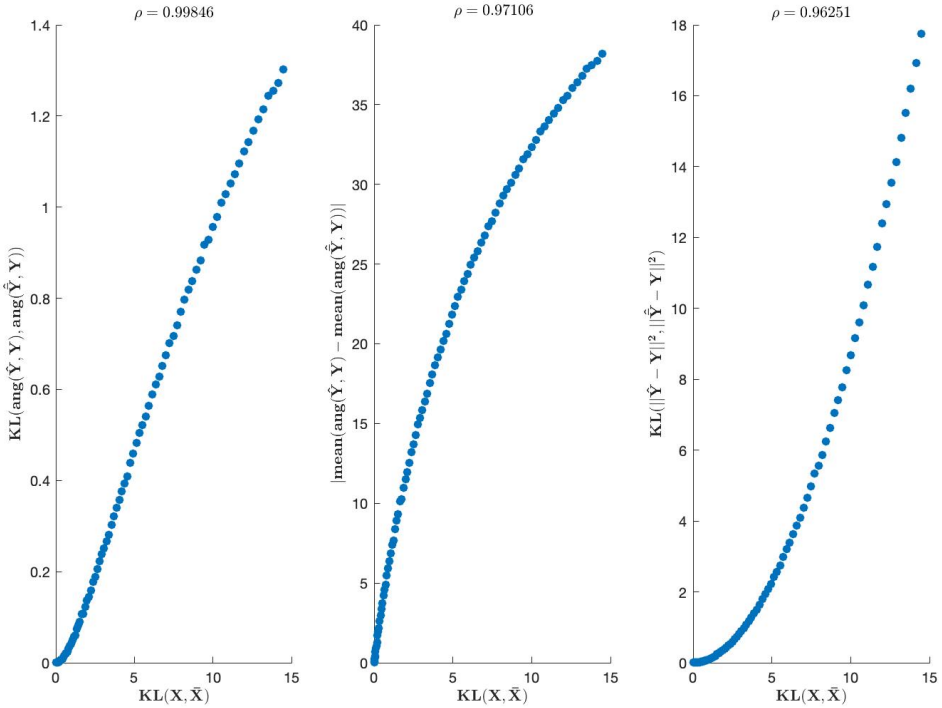


Figure 3.2: How distribution of neural data change with angle and error norm, when using the same decoder and while base changes. Simulation used.

2. **Noise Change** Similar to the procedure that was applied to obtain figures 3.1 and 3.2, we evaluated the Gaussian KL distances when the noise covariance matrix Σ of the original encoding model changes linearly, i.e., $\overline{\Sigma} = t_s\Sigma + (1 - t_s)\Sigma^*$, where t_s takes 90 equally-distanced values in $[0, 1]$, and Σ^* is a randomly chosen symmetric positive definite matrix.

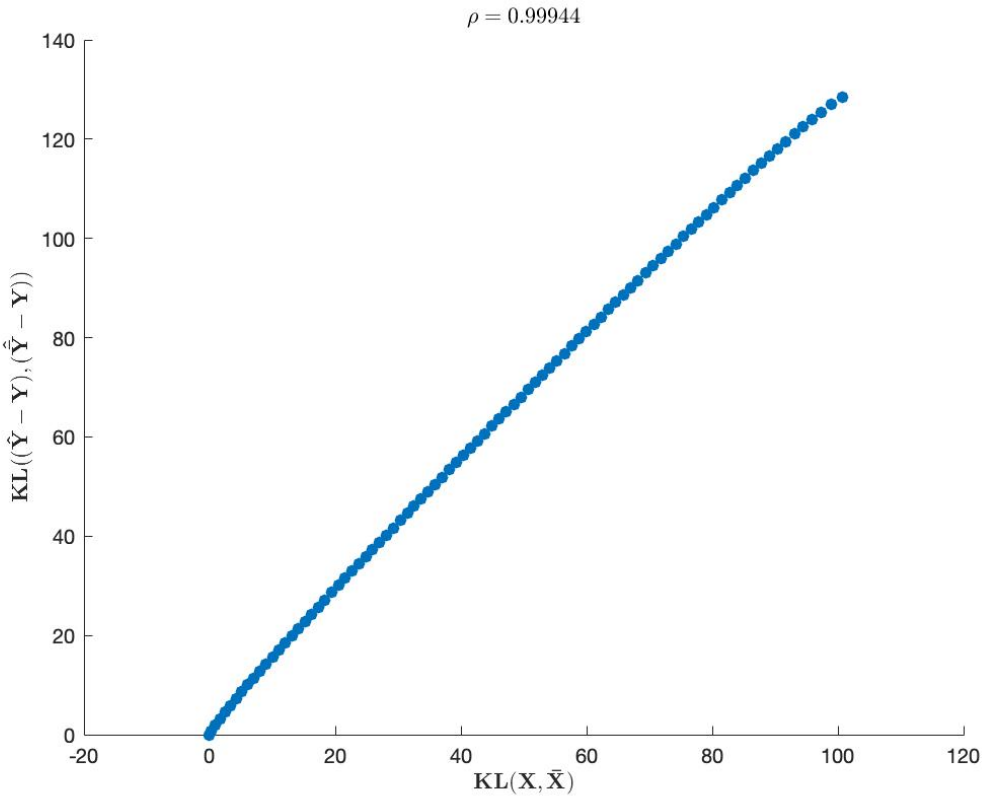


Figure 3.3: How distribution of neural data changes with distribution of error, when using the same decoder and while noise changes. No simulation

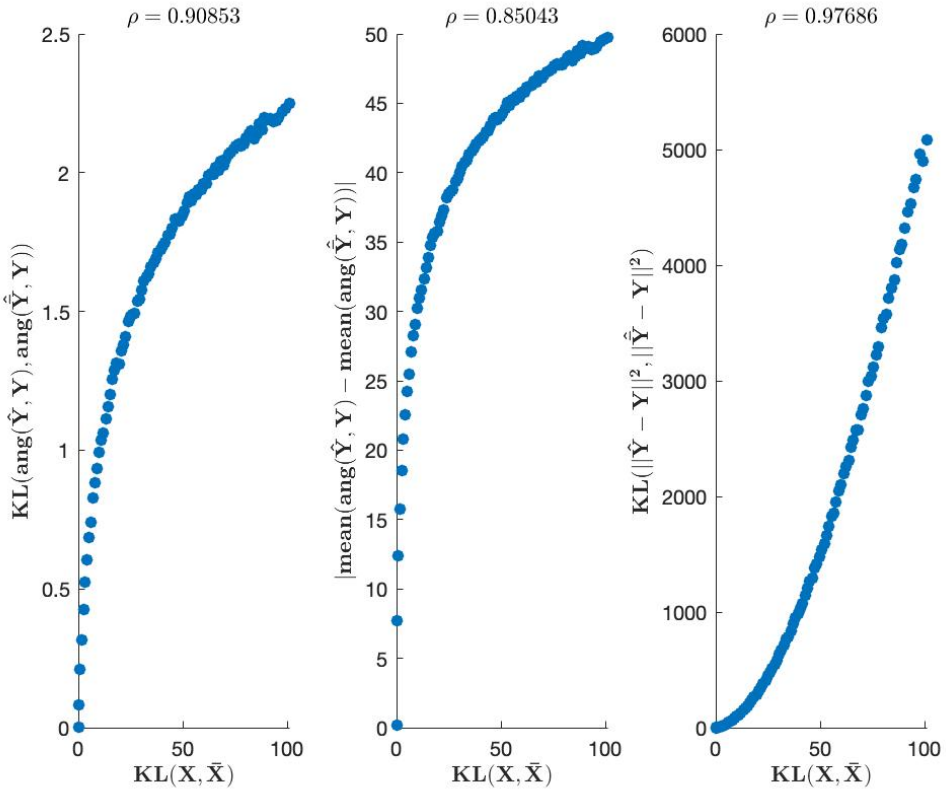


Figure 3.4: How distribution of neural data change with angle and error norm when using the same decoder and while noise changes. Simulation used.

3. **Rotation of Preferred Direction**

To explore how the rotation of preferred direction of neurons affect on the Gaussian KL distances, we first show the result for the perturbation paradigm in which the tuning of 5 neurons (out of 20) changes synchronously, i.e., $\overline{A_{<i>}} = A_{<i>} \begin{bmatrix} cos(t) & -sin(t) \\ sin(t) & cos(t) \end{bmatrix}$ where i is the index of each of those 5 neurons and $B_{<j>}$ represents the j'th row of matrix B (t takes 90 equally-distanced values in $[0, 0.9]$):

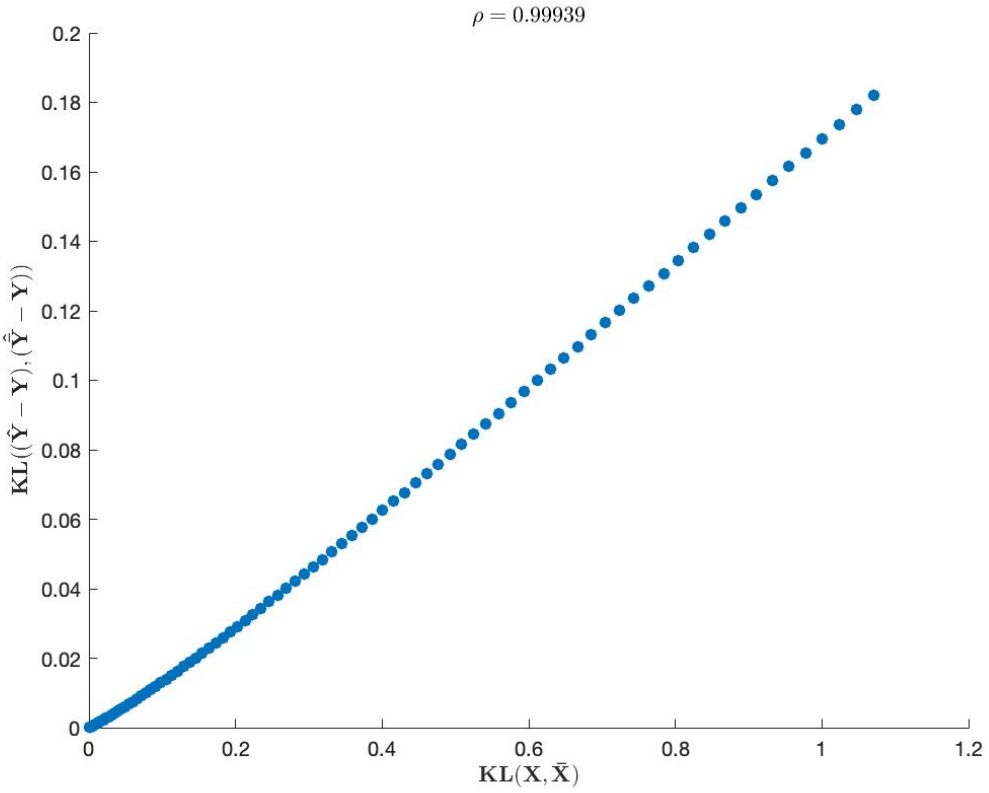


Figure 3.5: How distribution of neural data changes with distribution of error, when using the same decoder and while tuning of 5 neurons change. No simulation.

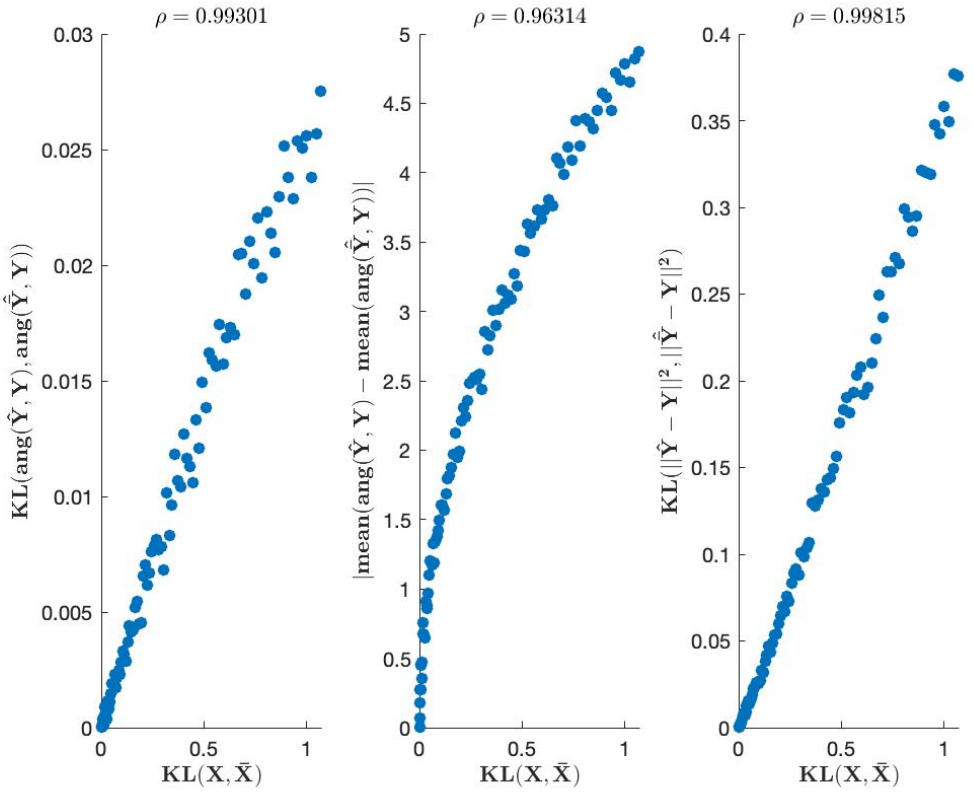


Figure 3.6: How distribution of neural data change with angle and error norm when using the same decoder and while tuning of 5 neurons change. Simulation used.

Increasing the number of neurons whose preferred direction is being rotated, we observed that the monotonic trend is still in place, unless we we change the tuning of all neurons by the same amount of rotation, i.e., $\overline{A} = A \begin{bmatrix} cos(t) & -sin(t) \\ sin(t) & cos(t) \end{bmatrix}$, for $t \in [0, 0.9]$. The results for this case are shown below:

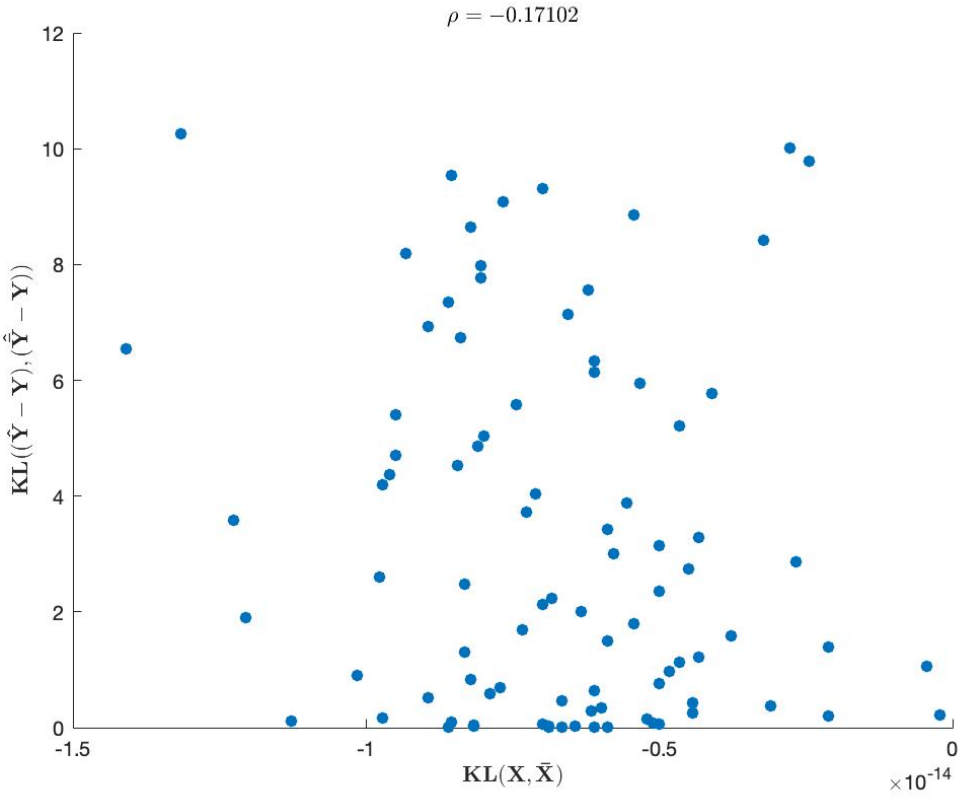


Figure 3.7: How distribution of neural data changes with distribution of error, when using the same decoder and while tuning of all neurons change. No simulation.

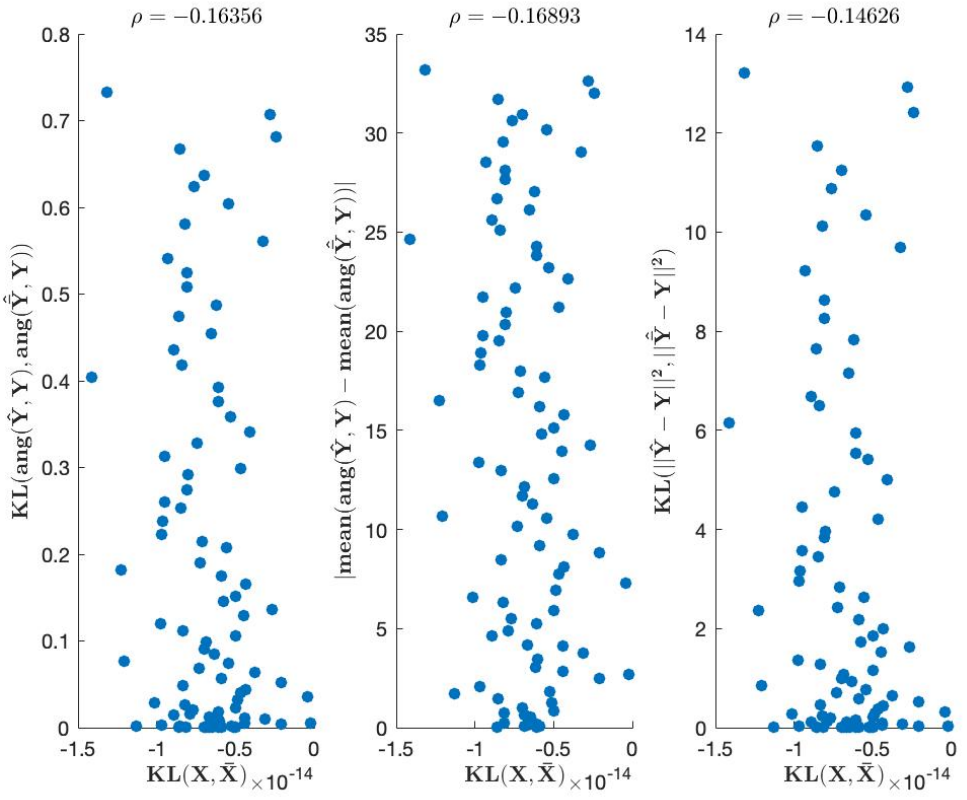


Figure 3.8: How distribution of neural data change with angle and error norm when using the same decoder and while tuning of all neurons change. Simulation used.

This phenomena roots from the rotation invariance of the distribution of Y. In fact, since $Y \sim N_l(\mathbf{0}, I)$, thereby having spherically symmetrical distribution, when we apply the rotation (by the same value) to all neurons, we maintain the distribution of X unchanged. In other words, if the original encoding model is $X = AY + b + Z$ as in (3.1), and R is the rotation matrix, the new encoding model turns to be $\overline{X} = ARY + b + Z$ as baseline and noise parameters has not changed. Now since the distribution of Y is invariant with respect to rotation, so RY is equally distributed as Y, which results in same distribution of X and $\overline{X}$. This can be confirmed from figures 3.7 and 3.8 that $KL(X, \overline{X})$ is almost zero for any rotation of all neurons. As a second confirmation, we can see from equation (3.15) that changing A to AR, that KL term turns to zero as R is an orthogonal matrix ($RR^\top = R^\top R = I$).

4. **Change of Amplitude**

Here we investigate the change the parameters of the original encoding model by perturbing the amplitude of the tuning function of neurons (the modulation depth). We set $\overline{A} = tA$, for t taking equally-distanced values in $[1, 2]$.

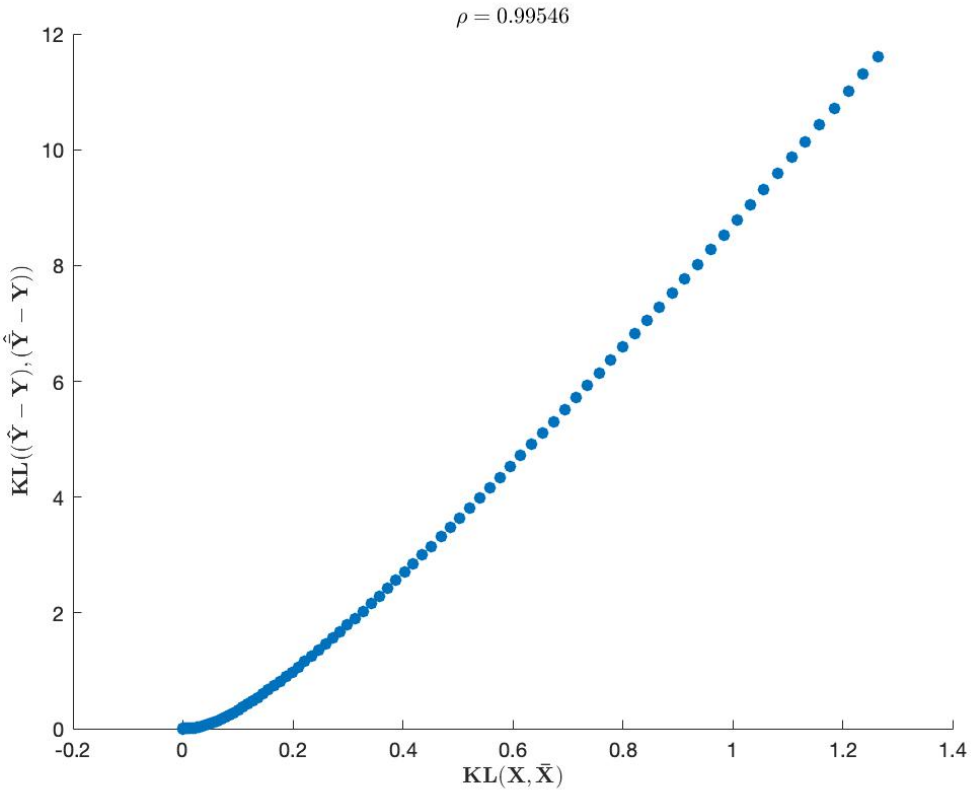


Figure 3.9: How distribution of neural data changes with distribution of error, when using the same decoder and while amplitude of all neurons change. No simulation.

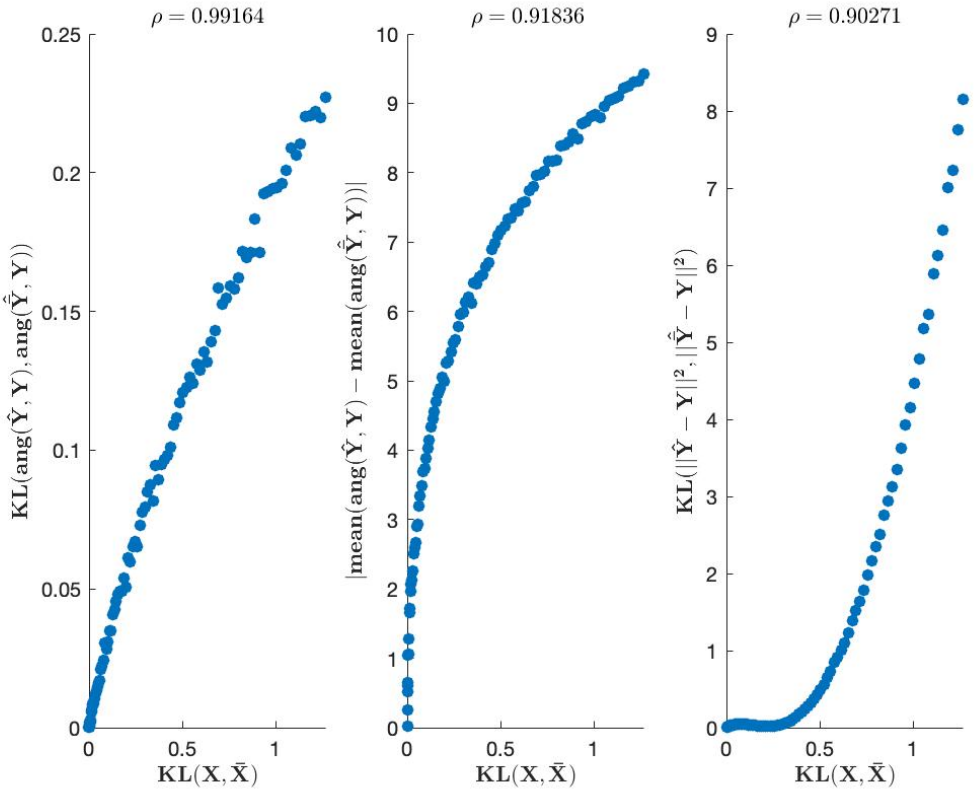


Figure 3.10: How distribution of neural data change with angle and error norm when using the same decoder and while amplitude of all neurons change. Simulation used.

Performing change of amplitude on subsets of the collection of 20 neurons, we observed the same monotonic trend as above.

5. **Drop of Channels**

To address the effect of channel removal on Gaussian KL distances, the new perturbation paradigm we define consist of two different ways: (I) at each step t ($1 \leq t \leq 20$), the t'th randomly chosen neuron is removed, i.e., the row of A indexed by that neuron is changed to a vector of zeros, as well as the associated element of vector b. In fact, that neuron is assumed to become silent. and (II) at each step t ($1 \leq t \leq 20$), the neuron with t'th lowest magnitude of tuning (i.e., the l^2-norm of the associated row in A) becomes silent, by the same fashion as in (I), namely, changing the associated row of A and b to zeros. In the former method we silent the neurons in a cumulative way, while in the latter we silent

them with substitution. figures 3.11 and 3.12 show the results of Gaussian KL distance comparison for the first method, and figures 3.13 and 3.14 represent the results for the second approach.

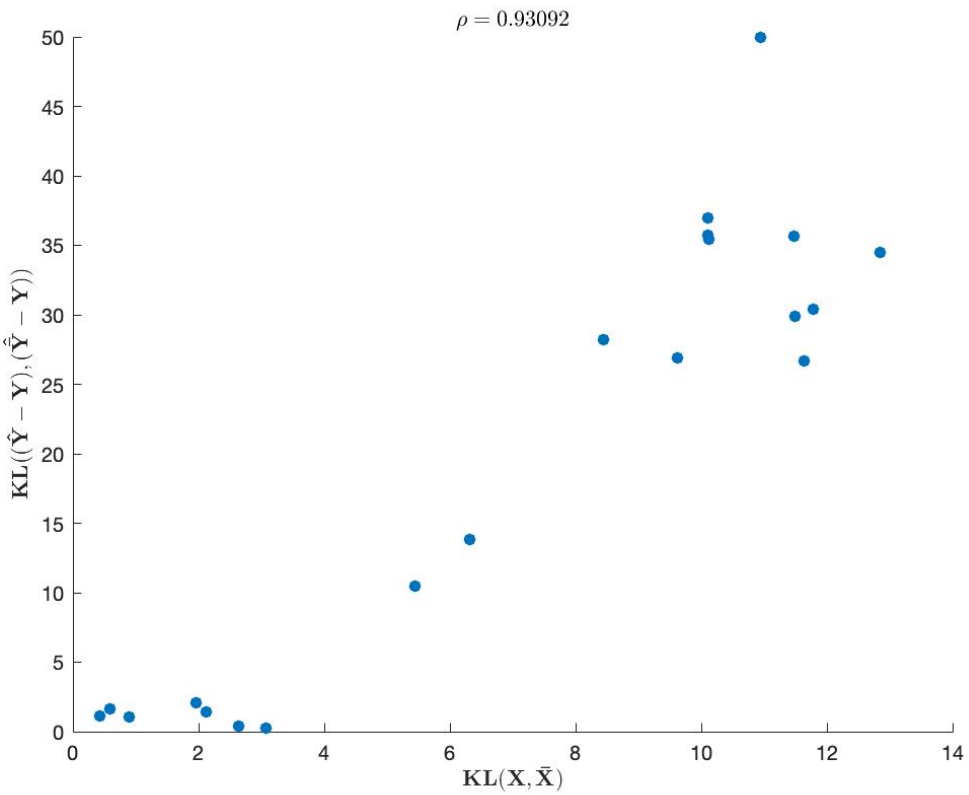


Figure 3.11: How distribution of neural data changes with distribution of error, when using the same decoder and while a new random neuron becomes silent at each step. No simulation.

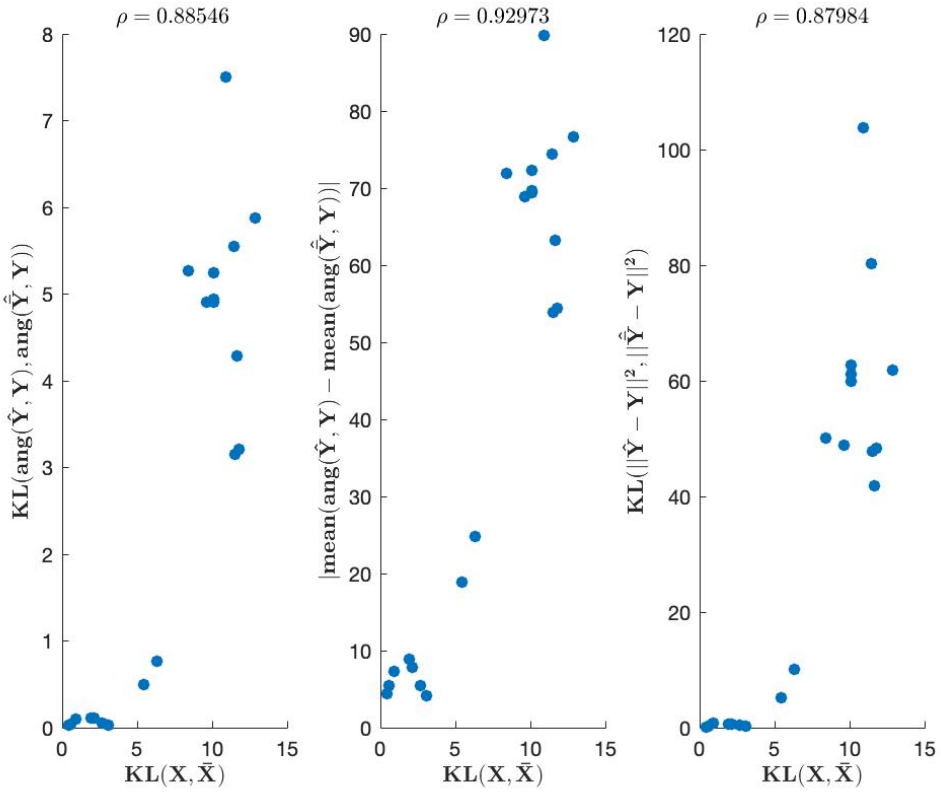


Figure 3.12: How distribution of neural data change with angle and error norm when using the same decoder and while a new random neuron becomes silent at each step. Simulation used.

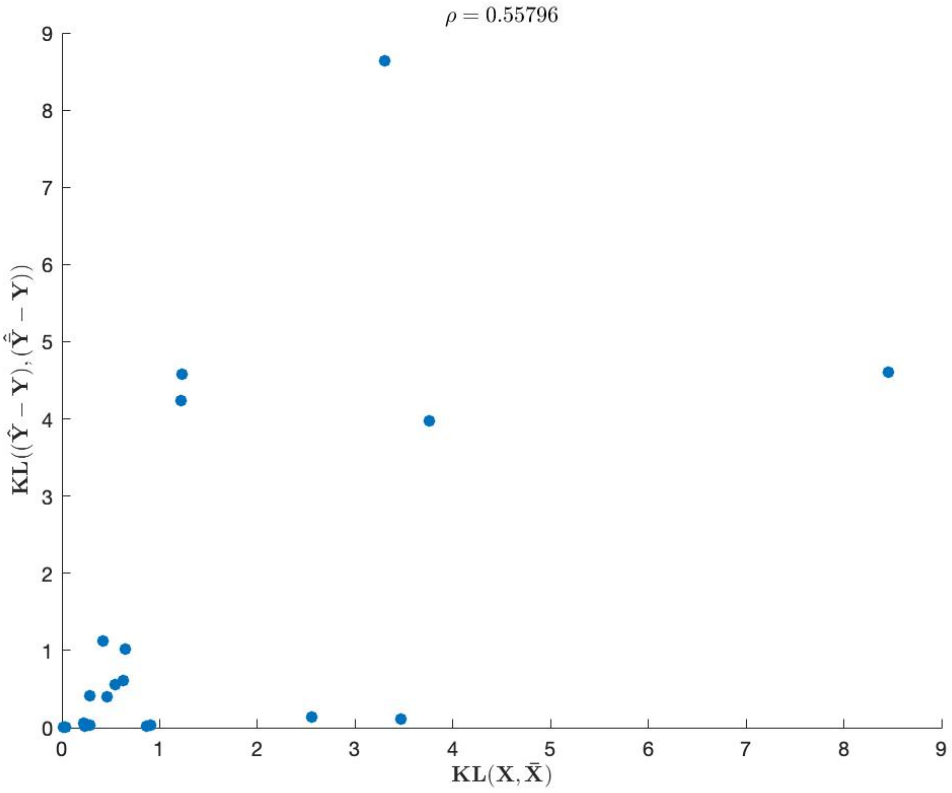


Figure 3.13: How distribution of neural data changes with distribution of error, when using the same decoder and while neurons becomes silent in order of least magnitude to the highest. No simulation.

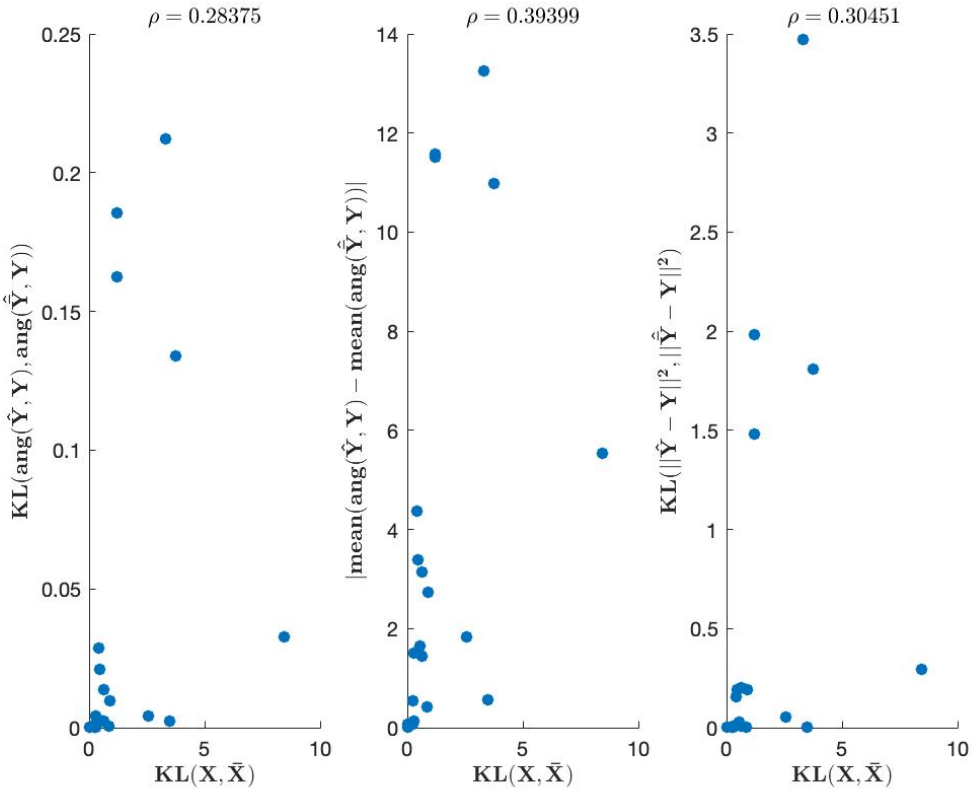


Figure 3.14: How distribution of neural data change with angle and error norm when using the same decoder and while neurons becomes silent in order of lowest magnitude to the highest. Simulation used.

3.2.4 Analysis of change of distribution of neural data given angular error in case of no model drift

In this section, we investigate the relation between the Gaussian KL distance of neural data given certain angular errors, when decoded using the optimal decoder found in section 3.2.2. The underlying probabilistic enconding model does not alter in this case, to capture the mere effect of instantaneous noise in performance error.

Figure 3.15 demonstrates the Gaussian KL distance between the distribution of neural data X and $X|\text{angular error} = \theta$. For both figures, θ changes from 0 to 180 degrees. Next, in Figure 3.16 we show the Gaussian KL distance between the conditional distribution of X given good performance (angular error of zero) and conditional distribution of X given other certain angular errors.

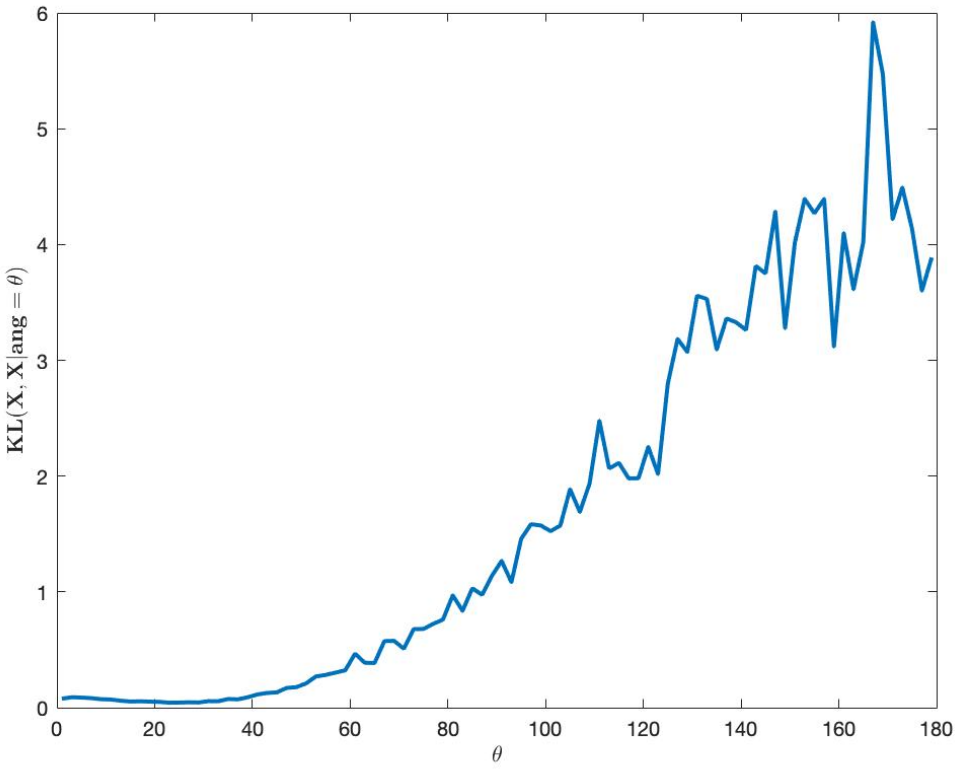


Figure 3.15: How far $P(X)$ is from $P(X|\text{angular error} = \theta)$ in Gaussian KL distance

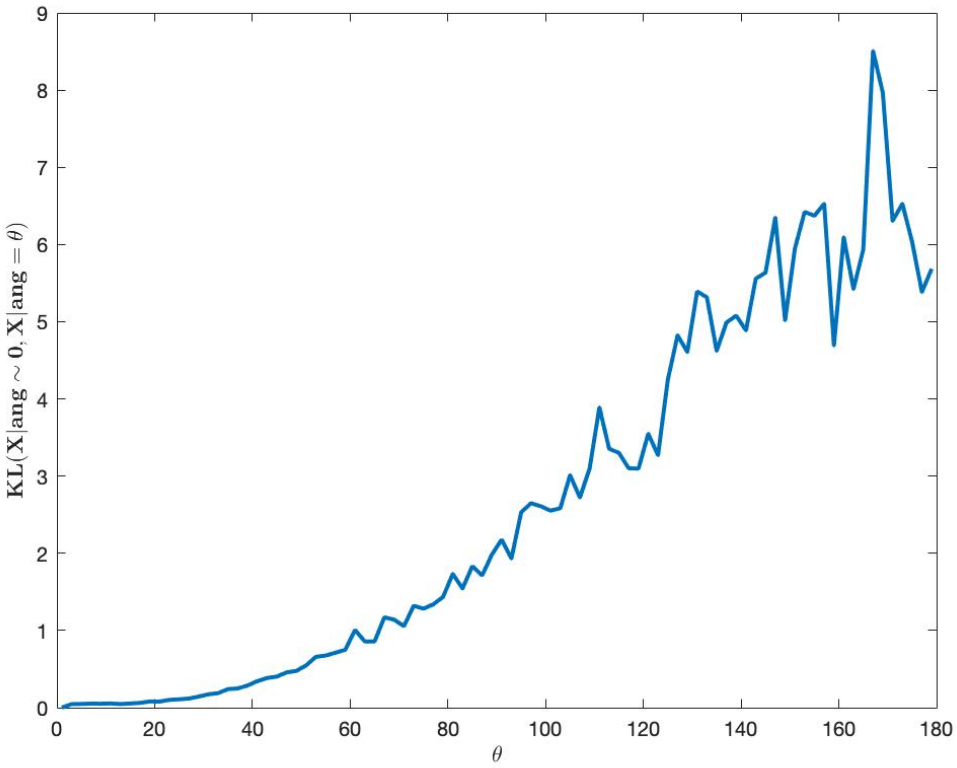


Figure 3.16: How far $P(X|\text{angular error} = 0)$ is from $P(X|\text{angular error} = \theta)$ in Gaussian KL distance

3.2.5 Comparative analysis between the contribution of noise and model drift on GKLD

A study by Pun et al. (2024) revealed a strong, linear correlation between the Gaussian Kullback-Leibler divergence (GKLD) and decoding performance [10]. This suggests that GKLD could be a useful metric for predicting performance. However, the type of error causing the GKLD increase plays a crucial role.

Angular error (AE) can arise from two main sources:

1. Transient Noise: This refers to short-lived variations in the neural signal, typically lasting less than a second. While it contributes to decoding errors, it doesn't cause persistent performance decline.

2. Model Drift: This describes gradual, long-lasting changes in the relationship

between neural activity and intended movements. Unlike transient noise, model drift can lead to a continuous decline in performance over time.

The distinction between these error types is crucial in clinical applications. If transient noise dominates the GKLD-performance relationship, combining neural activity from different time windows wouldn't improve performance prediction. This is because transient noise doesn't persist across time bins. However, if model drift is the primary cause, then analyzing consecutive time bins together might be informative. This is because these bins would likely share similar error characteristics due to the underlying drift.

To study how each noise and model drifts contributes to GKLD, This linear relationship can be recreated in simulation using noise or using model drift or using both:

Consider the following linear regression encoding model for neural data given kinematics,

$$X = AY + Z. \tag{3.17}$$

In this equation, X represents the matrix of neural features (size $d \times n$), Y is the movement intentions (size $l \times n$), and A denotes the regression coefficients (tuning) matrix (size $d \times l$). Here, we used $d = 50$, and $l = 2$. We assume a multidimensional Gaussian noise Z in this model, which is independent of Y and follows a distribution of $N_d(\mathbf{0}, \sigma^2 I)$.

For this linear model, under the assumption that $Y \sim N(\mathbf{0}, I)$, the optimal decoder $\mathbb{E}[Y|X]$ is $\widehat{Y} = BX$, where $B = A^T(AA^T + \sigma^2 I)^{-1}$. We denote $AE = \text{angle}(Y, \widehat{Y})$ as the angle error of this estimation.

We generate synthetic data by considering two scenarios:

Case (I): No added model drifts:

We simulate neural features using the encoding model in (3.1) for $n = 10^4$ timesteps

for 50 simulated sessions. Throughout these sessions, the parameters of the model A and σ^2 remain constant. This process yields the matrix $\overline{X}$, representing neural features of size $d \times 50n$. After decoding $\overline{X}$ using the optimal decoder $B = A^T(AA^T + \sigma^2 I)^{-1}$, we calculate the angle error for each time step across sessions.

Next, we estimated two conditional distributions:

1) The target distribution of neural features $\overline{X}$ conditioned on a specific angle error range $\theta - 2° < \overline{AE} < \theta + 2°$, and,

2) The reference conditional distribution of neural features $\overline{X}$ given good performance, where $0° < \overline{AE} < 4°$.

We compute the GKLD for varying angle errors $\theta \in \{2°, 6°, 10°, \cdots, 178°\}$.

Case (II): Model drifts with tuning changes is added (to mimic figure 1b by Pun et al. [10]):

The process remains akin to the one in case (I), except for a daily alteration in the parameter A of the encoding model. This parameter changes randomly each session day from the original tuning matrix A (used to build the decoder $B = A^T(AA^T + \sigma^2 I)^{-1}$) to a newly chosen tuning matrix A^*. This alteration is achieved using a linear combination of the original matrix and the new tuning matrix: $\lambda A + (1 - \lambda)A^*$, where λ is uniformly chosen at random from $[0, 1]$. Despite this daily change in the encoding model, the neural features $\overline{X}$ across all days is still decoded using the same decoder B, which was specifically optimized for $\lambda = 0$. (The other parameter of the encoding model (σ^2) remains unchanged daily). Similar to case (I), the GKLD between $p(\overline{X}|0° < \overline{AE} < 4°)$ and $p(\overline{X}|\theta - 2° < \overline{AE} < \theta + 2°)$ (both assumed Gaussian) for varying θ is calculated.

This procedure captures how the encoding model's parameter changes affect the distribution of neural features, even when decoded using an optimal decoder designed for a stable parameter ($\lambda = 0$). The GKLD is computed across different angle error ranges (θ) to assess the impact of these parameter variations. In contrast, in case (I), where there is no change in the encoding model parameters, the GKLD is solely

affected by the instantaneous noise present in the system. This distinction allows us to discern how variations in the encoding model parameters specifically contribute to the differences observed in the neural data distribution, separate from the impact of mere noise fluctuations.

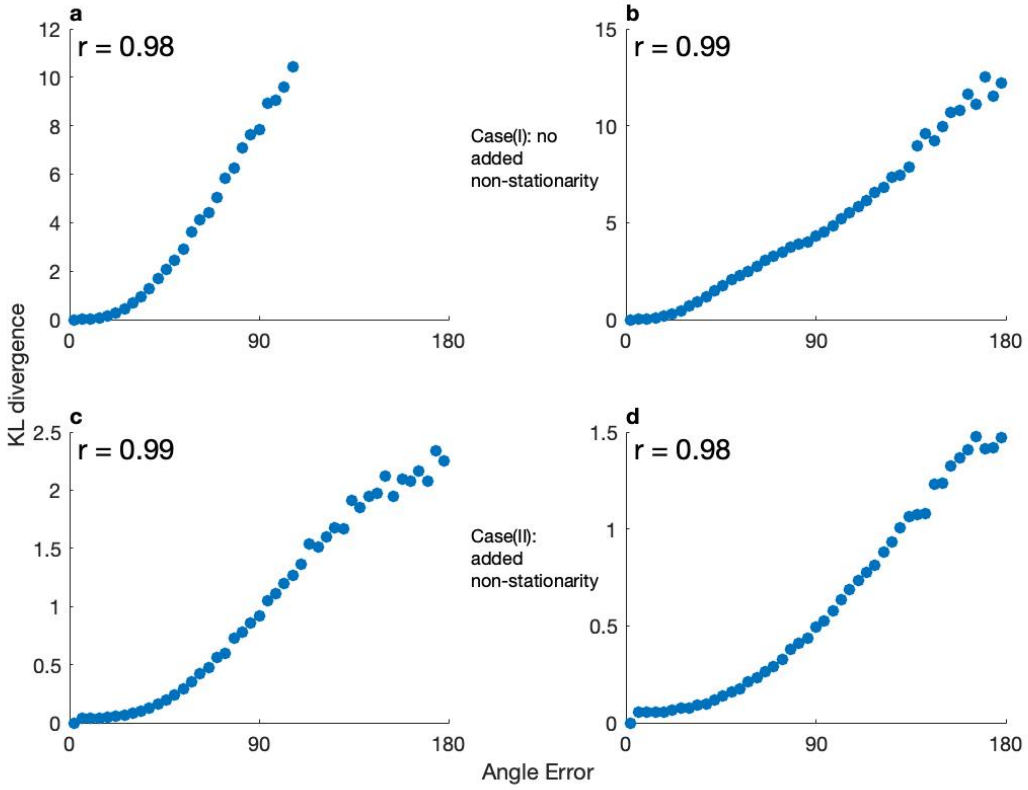


Figure 3.17: A linear relationship between GKLD and angle error in simulated noise or model drift. $KL(p(X|0 < AE < 4), p(X|\theta - 2 < AE < \theta + 2))$ changes with angle error θ. (a-b) displayed case (I) where noise was applied without additional model drifts, and (c-d) displayed case (II) where added model drifts (tuning changes) were applied. In (a), (c), we use a low model noise ($\sigma^2 = 4$), while in (b), (d), a higher model noise ($\sigma^2 = 25$) is applied. Pearson correlation coefficient was shown in each panel.

Both simulated noise and model drift result in a linear relationship between $KL(p(X|0 < AE < 4), p(X|\theta - 2 < AE < \theta + 2))$ with θ. Additionally, a higher level of noise corresponds to increased angle error. The difference in coverage of angle errors is notable: (c) spans the entire possible range (from 0° to 180°), while panel (a) only extends up to 110°. This discrepancy potentially underscores the contribution of model drifts to the GKLD, surpassing mere noise effects

3.3 Conclusion

The MINDFUL score proposed by Pun et al. [10] is based on their observation that the Gaussian Kullback-Leibler divergence (KLD) of neural data correlates strongly with angular error over time, as seen in two individuals with tetraplegia. In this chapter, we investigated this relationship within the framework of linear Gaussian encoding models.

Our findings indicate that specific types of model drift can indeed generate the observed linear relationship between KL divergence and angular error. However, other model perturbations may not exhibit this pattern. This distinction is important for determining when the MINDFUL score may be practically useful, and it offers insight into the types of model drift that may occur in the motor cortex.

Furthermore, we demonstrated that this linear relationship between Gaussian KLD and angular error can also arise independently of model drift, as a mere consequence of noise. However, when model drift is present, the relationship remains linear but with a greater range of angular error. This suggests that while MINDFUL can be a valuable tool, its interpretation requires careful consideration of both model drift and noise factors.

3.4 Appendix

Proof of Lemma 7:

Proof. Let the linear decoder minimizing the loss function of mean squared error $\mathbb{E}[||Y - g(X)||^2]$ be $\widehat{Y} = g(X) = PX + q$.

Step 1: We first find the optimal q for a fixed P:

$$\begin{aligned} q &= \arg\min_q \mathbb{E}[||Y - (PX + q)||^2] \\ &= \arg\min_q \mathbb{E}[||Y - PAY - Pb - q - PZ||^2] \\ &= \arg\min_q \mathbb{E}[||(I - PA)(Y - \mu) - PZ - (Pb + q - (I - PA)\mu)||^2] \\ &= \arg\min_q \mathbb{E}[||(I - PA)(Y - \mu) - PZ - h(q)||^2] \end{aligned} \tag{3.18}$$

where $h(q) = Pb + q - (I - PA)\mu$. We know that for any random vector Y with $\mathbb{E}[||Y||^2] < \infty$ we have:

$$\arg\min_{\overrightarrow{t}} \mathbb{E}[||Y - \overrightarrow{t}||^2] = \mathbb{E}[Y] \tag{3.19}$$

And since $\mathbb{E}[(I - PA)(Y - \mu) - PZ] = \mathbf{0}$, for any P, the optimal q where $q = \arg\min_q \mathbb{E}[||(I-PA)(Y-\mu)-PZ-h(q)||^2]$ is the root of $h(q)$, i.e., $q = (I-PA)\mu - Pb$. Hence, it is only remained to find the optimal P.

Step 2: Note that

$$\begin{aligned} P &= \arg\min_P \min_q (\mathbb{E}[||Y - (PX + q)||^2]) \\ &= \arg\min_P \mathbb{E}[||Y - PX - \big((I - PA)\mu - Pb\big)||^2] \\ &= \arg\min_P \mathbb{E}[||(Y - \mu) - P(AY + b + Z) + PA\mu + Pb||^2] \\ &= \arg\min_P \mathbb{E}[||(Y - \mu) - P\big(A(Y - \mu) + Z\big)\big)||^2] \end{aligned} \tag{3.20}$$

which means that to find the optimal P, it suffices to find it for centered Y and X, i.e., $X = AY + Z$ in the next step.

Step 3:

$$
\begin{aligned}
P &= \arg\min_P \mathbb{E}[||Y - PX||^2] \\
&= \arg\min_P \mathbb{E}[\mathrm{Tr}(Y - PX)(Y - PX)^\top] \\
&= \arg\min_P \mathbb{E}[\mathrm{Tr}(YY^\top - YX^\top P^\top - PXY^\top + PXX^\top P^\top)] \\
&= \arg\min_P \mathbb{E}[\mathrm{Tr}(YY^\top) - 2\,\mathrm{Tr}(YX^\top P^\top) + \mathrm{Tr}(PXX^\top P^\top)] \\
&= \arg\min_P \mathrm{Tr}(\mathbb{E}[YY^\top]) - 2\,\mathrm{Tr}(\mathbb{E}[YX^\top P^\top]) + \mathrm{Tr}(\mathbb{E}[PXX^\top P^\top]) \\
&= \arg\min_P \mathrm{Tr}(\Gamma) - 2\,\mathrm{Tr}(\mathbb{E}[YX^\top]P^\top) + \mathrm{Tr}(P\,\mathbb{E}[XX^\top]P^\top) \\
&= \arg\min_P \mathrm{Tr}(\Gamma) - 2\,\mathrm{Tr}(\mathbb{E}[YY^\top A^\top + YZ^\top]P^\top) \\
&\qquad + \mathrm{Tr}(P\,\mathbb{E}[AYY^\top A^\top + AYZ^\top + ZY^\top A^\top + ZZ^\top]P^\top) \\
&= \arg\min_P \mathrm{Tr}(\Gamma) - 2\,\mathrm{Tr}((\Gamma A^\top)P^\top) + \mathrm{Tr}(P(A\Gamma A^\top + \Sigma)P^\top) \\
&= \arg\min_P \mathrm{Tr}(\Gamma - 2\Gamma A^\top P^\top + PA\Gamma A^\top P^\top + P\Sigma P^\top) \\
&= \arg\min_P \mathrm{Tr}(\Gamma - 2P(A\Gamma A^\top + \Sigma)^{1/2}(A\Gamma A^\top + \Sigma)^{-1/2}A\Gamma^\top \\
&\qquad + P(A\Gamma A^\top + \Sigma)^{1/2}(A\Gamma A^\top + \Sigma)^{1/2}P^\top \\
&\qquad + \Gamma A^\top(A\Gamma A^\top + \Sigma)^{-1/2}(A\Gamma A^\top + \Sigma)^{-1/2}A\Gamma^\top \\
&\qquad - \Gamma A^\top(A\Gamma A^\top + \Sigma)^{-1}A\Gamma^\top)
\end{aligned}
\tag{3.21}
$$

Let $B = (A\Gamma A^\top + \Sigma)^{1/2}$, and note that $B = B^\top$, therefore:

$$
\begin{aligned}
P = \arg\min_P \mathrm{Tr}(&PBB^\top P^\top - 2PBB^{-1}A\Gamma^\top + \Gamma A^\top B^{-1}B^{-1^\top}A\Gamma^\top) \\
&- \mathrm{Tr}(\Gamma A^\top(AA^\top + \Sigma)^{-1}A\Gamma^\top - \Gamma)
\end{aligned}
\tag{3.22}
$$

As $\mathrm{Tr}(\Gamma A^\top (AA^\top + \Sigma)^{-1} A\Gamma^\top - \Gamma)$ does not depend on P, we have

$$\begin{aligned} P &= \arg\min_P \mathrm{Tr}((PB - \Gamma A^\top B^{-1})(PB - \Gamma A^\top B^{-1})^\top) \\ &= \arg\min_P ||PB - \Gamma A^\top B^{-1}||^2 \\ &= \Gamma A^\top B^{-2} \\ &= \Gamma A^\top (A\Gamma A^\top + \Sigma)^{-1} \end{aligned} \tag{3.23}$$

This completes the proof. □

CHAPTER 4

Mitigating gesture interference in brain-computer interfaces with task-robust decoding models

4.1 Introduction

Brain-computer interfaces (BCIs) have the potential to enable individuals with severe motor disabilities to restore their independence, by decoding their neural activity into actionable commands [29, 38, 49, 55–57, 79].

During the training phase of these decoders for computer cursor control tasks, explicit calibration sessions are typically required. In these sessions, neural signals are recorded simultaneously with the intended cursor velocity (movement) or alongside other motor imagery (MI) tasks, such as various gestures, including clicks [29, 80–82].

Recent progress in the field has enabled rapid point-and-click or click-and-drag which resulted in better control of computer cursors [33–39]. However, achieving widespread adoption as a reliable assistive platform requires further advancements of decoding performance in such cases of multiple tasks, especially since the capability to decode more types of MI actions is crucial for rehabilitation [83].

One of the challenges in achieving high performance in point-and-gesture BCIs is the interference between, for example, the decoded movement intention and the other motor imaginary intentions (e.g. gestures). In other words, current decoding techniques fail to separate information in the neural features effectively; resulting in unintended cursor movement while gesturing or unintended gesturing during cursor movement [35]. Similar contamination of one decoder by the intention of the other task has been observed in prosthetic reach-and-grasp actions [28, 40].

To mitigate this challenge in the context of point-and-click BCI, Dekleva et al. [35] proposed using a new approach for click decoding that is based on transient neural signals they observed at the onset and offset of a MI task.

In this paper, we introduce two methods aimed at mitigating interference during concurrent movement and gesture in point-and-gesture BCI applications. Both our proposed methods require two (potentially distinct) training datasets: one for the primary task (e.g. movement) and another for the secondary MI task (e.g. gesture). This scenario is common in current BCI point-and-gesture systems, where separate training sessions for each decoder have yielded two separate datasets of neural signals/kinematics and neural signals/gestures [27, 34]. However, during BCI use, both movement and gesture are intended concurrently by the user and require simultaneous decoding. This highlights a key challenge: the joint distribution of neural signals, movement, and gestures often differs between training and BCI use time due to the fact that during training, only the primary task (e.g. movement) was intended, whereas, during BCI use, both movement and gestures are intended concurrently.

Our first method trains the primary task decoder (e.g., movement decoder) differently by incorporating a regularization term into the cost function. This term penalizes statistical dependencies [84] between the decoded intention of the primary task and the other (interfering) MI tasks (e.g., gesture) while ensuring sufficient decoding accuracy. As we demonstrate in the results section, our method offers significantly improved robustness to changes in the interfering task. Moreover, while the solution is potentially suboptimal

compared to an ideal scenario with identical training and BCI use distributions of neural data and both tasks, this method of leveraging the secondary task calibration dataset for training the primary task decoder has significantly more predictive power than the case of only using the primary task calibration dataset.

Our second method tackles interference by finding a specific subspace within the neural data representation on which any linear decoder of the primary task (e.g. movement) is robust to any "anticipated" change of the joint distribution of neural signals, movement, and gestures, which is the potential variations of the user's secondary task (e.g. gestures) intention during BCI use. We achieve this by leveraging the dataset containing paired neural activity and the secondary task information. The neural data is then projected onto this subspace, effectively removing the "secondary task noise". This projected data can then be directly fed into an existing potentially non-linear decoder of the primary task. Crucially, this approach avoids the need for retraining the decoder, unlike the first method. It leverages the existing decoder's capabilities while ensuring any linearly decoded primary task remains uncorrelated with the secondary intentions, thereby mitigating interference.

The first approach shares similarities with the field of fairness in machine learning [85–87]. In fair regression, the researchers study the trade-off between the accuracy of a predictor of a real-valued function such as salary, and its "fairness" concerning demographic-sensitive characteristics like gender or race. A predictor satisfies the constraint of "Statistical (Demographic) Parity" if the predicted output is independent of the protected attribute [88–92]. In the context of brain-computer interfaces (BCIs), we can analogously treat the interfering secondary task as the protected attribute that varies during decoding, aiming to predict the primary task intention (akin to salary prediction) from neural signals (candidate skills) while ideally having the primary task prediction independent of the secondary MI intention (gender). Specifically, our first method draws on the work of Perez et al. [93], which employs the Hilbert-Schmidt Independence Criterion (HSIC) to quantify independence [84]. However, unlike studies in fair machine learning, there is typically limited or no training dataset combining multiple types of

kinematics and other MI task intentions simultaneously in the BCI context. Additionally, during BCI use, the user's intended secondary task is also unknown, i.e., both primary and secondary task intentions require real-time decoding solely from neural signals, while some research studies on fairness in machine learning assume access to the protected attribute at prediction time [94].

4.2 Methods

4.2.1 Regularized decoder training

Let $X = [X_1, X_2, \cdots, X_{d_X}] \in \mathbb{R}^{d_X}$, $Y = [Y_1, Y_2, \cdots, Y_{d_Y}] \in \mathbb{R}^{d_Y}$ and $Z = [Z_1, Z_2, \cdots, Z_{d_Z}] \in \mathbb{R}^{d_Z}$ represent the random vectors of neural data, user's intended primary task (e.g. movement), and user's intended secondary task (e.g. gestures) for a single timestep respectively, and $d_Y, d_Z < d_X$. We consider a scenario where existing datasets from movement and gesture calibration sessions are available. These datasets represent the relationship between neural activity and intended actions:

- Primary decoder calibration: The recorded neural signal during MI task calibration is denoted by $\mathbf{X} \in \mathbb{R}^{n \times d_X}$, where each row represents a single timestep's sample of the random vector X. The corresponding intended task features are stored by $\mathbf{Y} \in \mathbb{R}^{n \times d_Y}$. n represents the number of timesteps in calibration sessions of the task.

- Secondary decoder calibration: The recorded neural signal during the second task calibration session is denoted by $\mathbf{X'} \in \mathbb{R}^{n' \times d_X}$. The corresponding intended task features are represented by $\mathbf{Z'} \in \mathbb{R}^{n' \times d_Z}$. n' denotes the number of timesteps in calibration sessions.

Throughout this section paper, to focus on core concepts, we consider a scenario where the primary task to be decoded is movement, and the interfering secondary task is gestures. It is important to emphasize, however, that these roles are interchangeable. Therefore, our goal is to design a decoder that accurately translates neural signals into

intended movements while minimizing any statistical dependence on other MI task intentions. Our proposed approach to train a kinematics decoder is to find a function h_λ that attains:

$$h_\lambda = \underset{h \in \mathcal{C}}{\arg\min}\, \mathbb{E}[L(Y, h(X))] + \lambda \mathbb{D}[h(X'), Z'] \tag{4.1}$$

Here, the function h defines a decoder, i.e., $\widehat{Y} = h(X)$, and it is restricted to a class of functions $\mathcal{C}$. L represents a loss function to measure the discrepancy between the predicted and ground-truth kinematics. Additionally, $\mathbb{D}$ represents any measure of dependence between the decoded movementt (i.e., $h(X')$) and Z'.

The first term in the objective function in (4.1) measures the error of the prediction, while the second term quantifies the dependence between this decoder with the user's gesture intention. The non-negative hyperparameter λ controls the trade-off between performance on movement decoding and unwanted influence by other intended actions.

Linear decoding techniques have been extensively applied for kinematics decoding in BCI systems [28, 29, 35, 36]. Here, when $\mathcal{C}$ is restricted to linear functions, (4.1) can be rewritten as the following regularized optimization problem:

$$(A_\lambda, b_\lambda) \triangleq \underset{\substack{(A,b) \\ A \in \mathbb{R}^{d_X \times d_Y} \\ b \in \mathbb{R}^{1 \times d_Y}}}{\arg\min}\; \mathbb{E}[L(Y, XA + b))] + \lambda \mathbb{D}[X'A + b, Z'] \tag{4.2}$$

A measure of dependence can be defined as $\mathbb{D}[h(X'), Z'] = \|\text{cov}\left(h\left(X'\right), Z'\right)\|_F^2$, where $||.||_F$ represents the Frobenius norm [84], and cov denotes the cross-covariance matrix between the decoded movement $h(X')$ and Z'. Note that the cross-covariance of random vectors $R \in \mathbb{R}^{1\times n}$ and $W \in \mathbb{R}^{1\times m}$ is the $n \times m$ matrix defined as follows: $\text{cov}(R, W) \triangleq \mathbb{E}[(R - \mathbb{E}[R])^\top (W - \mathbb{E}[W])]$.

If 1) $\mathbb{D}[h(X'), Z']$ is as above, 2) the loss function L is the squared error $L(Y, \widehat{Y}) = \|Y - \widehat{Y}\|^2$ and 3) Σ_X is positive-definite, then the closed-form solution of (4.2) for $\lambda \geq 0$ is as follows (see supplementary):

$$
\begin{aligned}
A_\lambda &= \left(\Sigma_X + \lambda \Sigma_{X'Z'} \Sigma_{Z'X'}\right)^{-1} \Sigma_{XY} \\
b_\lambda &= \mu_Y - \mu_X A_\lambda
\end{aligned}
\tag{4.3}
$$

where $\Sigma_W = \text{cov}(W, W)$, $\Sigma_{WR} = \text{cov}(W, R)$, and $\mu_W = \mathbb{E}[W]$ for row random vectors W and R. For the derivation above and throughout the paper, we assume that covariance matrices Σ_X, Σ_Y, $\Sigma_{X'}$, and $\Sigma_{Z'}$ are finite-valued.

Remark 2. In (4.2), $\lambda = 0$ corresponds to the optimal linear estimator (OLE). If additionally (X, Y) is distributed as multivariate Gaussian, then the optimal decoder that minimizes MSE loss among all (possibly non-linear) estimators is OLE, i.e., $\mathbb{E}[Y|X] = XA_0 + b_0$.

Remark 3. Linear Gaussian encoding model is widely used for encoding neural activity given kinematics [10, 26–30]. It is described as below:

$$
X|Y \sim N_{d_X}(YU + c, \Sigma) \tag{4.4}
$$

where $U \in \mathbb{R}^{d_Y \times d_X}$, $c \in \mathbb{R}^{1 \times d_X}$ is the baseline firing rate [17], and the additive zero-mean Gaussian noise (distributed as $N_{d_X}(0, \Sigma)$) is assumed independent of Y.

Then it is important to note that under such encoding model assumption, and an additional assumption of multivariate Gaussian distribution for Y, (X, Y) will be multivariate Gaussian, and consequently, as stated in Remark 2, the optimal (possibly non-linear) predictor is $h(X) = XA_0 + b_0$ (OLE).

Under the encoding model described in Remark 3, the optimal linear estimator (OLE) would have been the optimal decoder in case of no distribution shift between movement calibration time and the BCI use time. However, with a distribution shift that is caused by the introduction of new MI tasks during BCI use, OLE will not necessarily remain optimal. This is why, in the regularization term in (4.1), the side dataset $(\mathbf{X}', \mathbf{Z}')$ is leveraged to train the decoder to be invariant to the distribution changes brought by their anticipated source, which is gesture intention.

Remark 4. Apart from linear decoders, we investigated more general decoder functions represented by $\widehat{Y} = f(XA + b, A, b)$. Note that in (4.2), $f(XA + b, A, b) = XA + b$. For instance, we define

$$f(XA + b, A, b) \triangleq \arg\min_{y} \mathbb{E}[L(Y, y)|XA + b] \tag{4.5}$$

When the loss L is the squared error loss, the defined decoder is $f(XA + b, A, b) = \mathbb{E}[Y|XA + b]$. Also, when L is 0-1 loss function ($L(Y, \widehat{Y}) = \mathbb{1}_{Y \neq \widehat{Y}}$) which can be used for discrete Y, e.g., when the primary task Y to be decoded is gestures and the secondary task Z is movement, f can be written as $f(XA + b, A, b) = \text{argmax}_y \mathbb{P}(Y = y|XA + b)$.

For any function f, we define $(A^f_\lambda, b^f_\lambda)$ as below:

$$(A^f_\lambda, b^f_\lambda) \triangleq \arg\min_{\substack{(A,b) \\ \|A\|=1}} \mathbb{E}\big[L\big(Y, f(XA + b, A, b)\big)\big] + \lambda \|\text{cov}(X'A, Z')\|^2_F \tag{4.6}$$

This formulation aims to find a projection of neural data that contains sufficient information to accurately decode Y, while penalizing projections that contain information about Z'. For the case of discrete Y, the loss L is considered a 0-1 loss function ($L(Y, \widehat{Y}) = \mathbb{1}_{Y \neq \widehat{Y}}$). See supplementary material for details.

4.2.1.1 Hyper-parameter tuning

Regularization is a common technique used to prevent overfitting in machine learning models. In this work, however, we leverage it to address a more fundamental challenge: distribution shift. This occurs when the underlying probability distributions governing the neural features, movements, and gesture intentions differ between the training data and the real-world use case (test data). This distinction is important because traditional cross-validation techniques, which rely on splitting the training data, are no longer suitable for hyperparameter tuning in this case. To address this challenge, we employ a data-driven heuristic approach to select a hyperparameter for decoder training. This approach is described in the steps below.

1. **Encoding model assumptions**: We begin by assuming linear Gaussian encoding models for both calibration datasets (cf. Remark 3):

$$\begin{aligned} X &= YU + \epsilon \quad \text{for } (\mathbf{X}, \mathbf{Y}) \\ X' &= Z'V + \epsilon' \quad \text{for } (\mathbf{X}', \mathbf{Z}') \end{aligned} \tag{4.7}$$

where $\epsilon \sim N_{d_X}(\mathbf{0}, \Sigma_\epsilon)$ and $\epsilon' \sim N_{d_X}(\mathbf{0}, \Sigma_{\epsilon'})$ are independent from Y and Z' respectively.

2. **Parameter estimation**: We compute the maximum likelihood estimator (MLE) of the encoding matrices (U and V) and noise covariances (Σ_ϵ and $\Sigma_{\epsilon'}$) from the training datasets $(\mathbf{X}, \mathbf{Y})$ and $(\mathbf{X}', \mathbf{Z}')$.

3. **Hypothetical test distribution**: We assume a hypothetical test distribution using the following model:

$$\bar{X} = \mu_X + (Y - \mu_Y)\widehat{U} + (Z' - \mu_{Z'})\widehat{V} + \delta \tag{4.8}$$

Here, $\bar{X}$ represents the hypothetical test neural data, and δ represents the noise random vector that is drawn from a multivariate Gaussian distribution $N_{d_X}(\mathbf{0}, \Sigma_\delta)$, where $\Sigma_\delta = \frac{1}{2}(\widehat{\Sigma_\epsilon} + \widehat{\Sigma_{\epsilon'}})$. Note that $\widehat{U}$, $\widehat{V}$, $\widehat{\Sigma_\epsilon}$, and $\widehat{\Sigma_{\epsilon'}}$ respectively represent the estimations of parameters U, V, Σ_ϵ, and $\Sigma_{\epsilon'}$ found in step 2.

This test distribution aids in hyperparameter tuning for decoder training, i.e., the optimal hyperparameter λ^* is determined as:

$$\lambda^* = \arg\min_{\lambda \geq 0} \mathbb{E}[\|Y - (\bar{X}A_\lambda + b_\lambda)\|^2] \tag{4.9}$$

Here, A_λ and b_λ represent the solution to the regularized objective function introduced in (4.2). (See equation (4.3) for their analytical form). In fact, (4.9) can be rewritten as below:

$$\lambda^* = \arg\min_{\lambda \geq 0} \operatorname{Tr}\left(\Sigma_{YX} M_\lambda \Sigma_{\bar{X}} M_\lambda \Sigma_{XY} - 2\Sigma_{YX} M_\lambda \Sigma_{\bar{X}Y}\right) \tag{4.10}$$

where $M_\lambda = (\Sigma_X + \lambda\Sigma_{X'Z'}\Sigma_{Z'X'})^{-1}$, $\Sigma_{\bar{X}} = \widehat{U}^\top\Sigma_Y\widehat{U} + \widehat{V}^\top\Sigma_{Z'}\widehat{V} + \Sigma_\delta$, and $\Sigma_{\bar{X}Y} = \widehat{U}^\top\Sigma_Y$. Note that Σ_X, Σ_Y, Σ_{YX}, $\Sigma_{Z'}$, and $\Sigma_{Z'X'}$ are the sample covariance matrices, estimated from the training datasets $(\mathbf{X}, \mathbf{Y})$ and $(\mathbf{X}', \mathbf{Z}')$. See supplementary (section 4.6.4.1) for details.

Remark 5. The simulated test data $\bar{X}$ with the additive model described above provides a controlled environment for the purpose of hyper-parameter tuning. However, to assess the effectiveness of this heuristic approach, we evaluated the predictive power of the decoder defined by A_{λ^*} and b_{λ^*} in more realistic scenarios for test data. In these scenarios, the test data is now modeled by $\tilde{X} = \mu_X + (Y - \mu_Y)\widehat{U} + s_Z(Z' - \mu_{Z'})\widehat{V} + s_\delta\delta$, where s_Z and s_δ are scaling factors for the influence of Z' and the simulated noise term $\delta \sim \frac{1}{2}(\widehat{\Sigma_\epsilon} + \widehat{\Sigma_{\epsilon'}})$, respectively. We compared the prediction power of $(A_{\lambda^*}, b_{\lambda^*})$ on these more realistic test data, with $(A_{\lambda_{min}}, b_{\lambda_{min}})$ where $\lambda_{min} = \arg\min_{\lambda\geq 0}\mathbb{E}[\|Y - (\tilde{X}A_\lambda + b_\lambda)\|^2]$ (see section (I)).

The results (see supplementary material) demonstrate that the performance of A_{λ^*} is sufficiently close to the optimal hyperparameter λ_{min} for varying s_Z and s_δ. This suggests that the heuristic approach using the simulated test data $\bar{X}$ generalizes well to more generic cases.

4.2.2 Interference-free decoding

In this section, we explore a novel strategy that allows the utilization of existing BCI decoders on transformed neural data. This transformation ensures that any linearly decoded kinematics becomes uncorrelated with gesture intentions. One might hope that this approach will also benefit non-linear decoders by reducing the influence of gestures on the movement decoding process. This might lead to increased robustness of movement decoding in real-world BCI scenarios where users might exhibit variations in gestures.

4.2.2.1 Neural transformation

Lemma 11. *Let $\Sigma_{Z'X'} \triangleq \text{cov}(Z', X')$ be well-defined and finite-valued. The following are equivalent properties of a matrix H (necessarily with d_X rows):*

(a) A satisfies $\|cov(X'A, Z')\|_F^2 = 0$ if and only if $A = HB$ for some B.

(b) The column space of H is equal to the null space of $\Sigma_{Z'X'}$.

If F is a $d_X \times d_X$ symmetric positive-definite matrix and $\Sigma_{Z'X'}F\Sigma_{X'Z'}$ is invertible, then

$$H = F - F\Sigma_{X'Z'}(\Sigma_{Z'X'}F\Sigma_{X'Z'})^{-1}\Sigma_{Z'X'}F$$

satisfies (a) and (b).

From this lemma (see supplementary for proof), we see that any linear decoder B applied on the transformed neural data under $T(X') = X'H$ will become uncorrelated with gesture intention Z'. We can estimate H which satisfies the assumption in lemma 11 by leveraging the available dataset $(\mathbf{X}', \mathbf{Z}')$, and employing the sample covariance matrix $\Sigma_{\mathbf{Z}'\mathbf{X}'}$.

4.2.2.2 Optimal linear estimator after transformation

We consider the following optimization problem :

$$\underset{\substack{(A,b) \\ \|\text{cov}(X'A,Z')\|_F^2=0}}{\arg\min} \quad \mathbb{E}[\|Y - (XA + b)\|^2] \tag{4.11}$$

We present a key theorem that characterizes the solutions to this optimization problem. In part (c) of this theorem we show that under the conditions that both $\Sigma_{Z'X'}$ and Σ_X are both of full rank, the optimization problem has a unique solution. We explore two additional cases in Corollaries 13 and 14 in supplementary, where only one of those two conditions holds.

Theorem 12. *Let Σ_X, Σ_Y, $\Sigma_{X'}$, and $\Sigma_{Z'}$ be well-defined and finite-valued.*

(a) Let $H \in \mathbb{R}^{d_X \times t}$ be any matrix with column space equal to null$(\Sigma_{Z'X'})$. *If $(B^*, c^*) \in \mathbb{R}^{t \times d_Y} \times \mathbb{R}^{1 \times d_Y}$ minimizes $\mathbb{E}[\|Y - (XHB + c)\|^2]$ over (B, c), then $(A^*, b^*) = (HB^*, c^*)$ is a solution to* (4.11).

(b) If $A_\infty = \lim_{\lambda \to \infty} A_\lambda$ and $b_\infty = \lim_{\lambda \to \infty} b_\lambda$ exist, then (A_∞, b_∞) is a solution to (4.11).

(c) If $\Sigma_{Z'X'}$ and Σ_X are both of full rank, then (4.11) *has the following unique closed-form solution*

$$\begin{aligned} A_\infty &= \Sigma_X^{-1} \left(I - \Sigma_{X'Z'} (\Sigma_{Z'X'} \Sigma_X^{-1} \Sigma_{X'Z'})^{-1} \Sigma_{Z'X'} \Sigma_X^{-1} \right) \Sigma_{XY} \\ b_\infty &= \mu_Y - \mu_X A_\infty \end{aligned} \tag{4.12}$$

See supplementary for the proof.

Remark 6. While our focus has been on linear decoders, the transformation T can also benefit non-linear decoders. One may hope that by applying any existing non-linear decoder function h to the transformed data $T(X)$, the decoded movement $\widehat{Y} = h(T(X))$ exhibits robustness to gesture intentions. In the scenario of a multivariate Gaussian joint distribution underlying X and Z, this robustness is inherently guaranteed. This guarantee arises from the observation that, in such cases, zero correlation between $T(X)$ and Z implies their independence. Consequently, $h(T(X))$ and Z are also independent.

4.3 Related work

Previous work by Wodlinger et al. and Collinger et al. [28, 29] employed a decoder that combines the Optimal Linear Estimator (OLE) with Ridge Regression. Ridge regression employs a regularization term ($\|A\|^2 = \mathrm{Tr}(A^\top A)$) to prevent overfitting. In contrast, our proposed regularization term (section 4.2.1) is $\|\mathrm{cov}\,(X'A, Z')\,\|_F^2 = \mathrm{Tr}\left((\Sigma_{Z'X'}A)^\top(\Sigma_{Z'X'}A)\right)$ (see supplementary), which is l_2-regularization after projecting A into a new direction defined by the covariance between Z' and X' from the side

dataset $(\mathbf{X}', \mathbf{Z}')$, and it is a data-dependent regularization designed to make the decoder less sensitive to variations in the secondary MI task.

Our work may be framed as a form of homogeneous-feature multi-task learning (MTL) [95]. It simultaneously predicts both classification (gesture) and regression (movement) tasks, thereby also qualifying as heterogeneous MTL as defined in [96]. Traditional MTL methods heavily rely on exploiting known or presumed relationships among tasks [97]. For example, in Alamgir [98] and Panagopoulos [99] employed MTL to leverage the shared information across subjects (each task corresponds to a subject), in order to increase the cross-subject efficiency of a MI task classifier [97], or in general, increase the generalization capability of the system [99]. In contrast, our approach does not use shared information across tasks. Additionally, in some MTL methods like the one proposed in [100], a regularization term is used that encourages orthogonality between the subspace spanned by the primary task and the subspace spanned by the unrelated secondary task. More specifically, if $\widehat{Z} = XB_{d_X \times d_Z}$ denotes the secondary task decoder, their objective function (to be minimized over (A, B)) can be written as below:

$$\mathbb{E}[\|XA - Y\|^2] + \mathbb{E}[\|XB - Z\|^2] + \lambda \|[A, B]\|_{2,1}^2 + \alpha \|A^\top B\|_F^2 \tag{4.13}$$

Here, the first regularization term is the $l_{2,1}$-norm of concatenated A and B, and it is the standard MTL penalty [101–105] to leverage shared features between tasks. The second regularization term (the squared Frobenius norm of $A^\top B$) encourages orthogonality between decoders of unrelated tasks.

However, in our study, the decoders of gesture and movement (A and B) are trained separately, and they are not imposed to be orthogonal.

The challenge of deviation between training and testing distributions has also been studied in curriculum learning (CL) [106, 107]. However, the nature of the distribution shift differs between our work and typical CL approaches. In curriculum learning, as discussed in [108–110], the shift often arises from noisy or weak annotations within the massive training data. In this approach, these corrupted low-confidence examples

are typically included in the later stages of the curriculum in CL framework to improve robustness. In [111], this CL approach is proposed to address distribution shifts caused by inter-individual differences to improve the robustness of cross-subject BCIs. In contrast, the discrepancy between test and train distribution in our study stems from the absence of one task during calibration and its anticipated subsequent presence and variation during use.

4.4 Results

In this section, we evaluate the performance of the proposed approaches in this study (Section 4.2). We employ simulated datasets to comprehensively examine the decoders' predictive power under various conditions.

To evaluate the performance of different decoders described in the methods section across various test scenarios, we employ a relative version of the expected loss function, which is $\frac{\mathbb{E}[\|Y-\widehat{Y}\|^2]}{\mathrm{Tr}(\mathrm{cov}(Y))}$. This normalization has a key advantage: If we estimate the movement variable (Y) by its mean (μ_Y), the resulting relative MSE becomes 1. Consequently, any decoder with a relative MSE exceeding 1 would be deemed ineffective. See supplementary (section 4.6.3) for details.

4.4.1 Subspace regimes in encoding models

As introduced in (4.7), the linear-Gaussian encoding models we assume for calibration datasets are:

$$X = YU + \epsilon \quad \text{for } (\mathbf{X}, \mathbf{Y})$$
$$X' = Z'V + \epsilon' \quad \text{for } (\mathbf{X}', \mathbf{Z}')$$

Here, each entry in U is derived independently from standard normal distribution, and each row is normalized to represent a unit vector.

This section first explores three distinct regimes for the subspaces $U \in \mathbb{R}^{d_Y \times d_X}$ and

$V \in \mathbb{R}^{d_Z \times d_X}$ in the encoding models in (4.7). We assume $\mu_Y = 0$, $\mu_Z = 0$ for simplicity. Fixing all parameters underlying the joint distribution of both training datasets, including Σ_Y, $\Sigma_{Z'}$, Σ_ϵ, $\Sigma_{\epsilon'}$ and the subspace U, we explore different scenarios for V:

1. **Overlapping subspaces $\mathbf{V} = \mathbf{V_0}$** : In this case, the first $\min(d_Z, d_Y)$ rows of U and V are identical. This represents a scenario where the subspaces for movement and gesture decoding overlap.

2. **Orthogonal subspaces $\mathbf{V} = \mathbf{V_1}$**: Here, V_1 is chosen randomly and independently from U. For sufficiently large dimensions d_X, the law of large numbers suggests that these subspaces are nearly orthogonal.

3. **Intermediate subspaces $\mathbf{V} = \mathbf{V}_\alpha$**: This case represents a spectrum between the previous two extremes. Here, $V_\alpha = \sqrt{1-\alpha}V_0 + \sqrt{\alpha}V_1$ for $0 < \alpha < 1$.

In any case, each row of V is normalized to represent a unit vector.

We consider a typical BCI use encoding model that incorporates both concurrent movement and gesture intention:

$$X^{test} = Y^{test}U + Z^{test}V + \epsilon^{test} \tag{4.14}$$

Here, Y^{test}, Z^{test}, and ϵ^{test} are assumed independent, and $\epsilon^{test} \sim N_{d_X}(\mathbf{0}, \Sigma_{\epsilon^{test}})$. Assuming a data-rich environment, we treat the parameters $\Sigma_{Y^{test}}$, $\Sigma_{Z^{test}}$ and $\Sigma_{\epsilon^{test}}$ as known. In other words, for $d_X = 300$, $d_Y = 2$, and $d_Z = 4$, we assume $\Sigma_{Y^{test}} = \Sigma_Y = I_{d_Y}$, and $\Sigma_{Z^{test}} = \Sigma_{Z'} = 1.5I_{d_Z}$ (test data covariance matches training data covariance for movement and gesture). The noise covariance of calibration datasets are assumed $\Sigma_\epsilon = \Sigma_{\epsilon'} = 0.1I_{d_X}$, and for test, we assume that the noise covariance $\Sigma_{\epsilon^{test}}$ is chosen such that the total "noise" variance remains constant across training and testing: $\mathrm{Tr}(V_\alpha^\top \Sigma_{Z^{test}} V_\alpha + \Sigma_{\epsilon^{test}}) = \mathrm{Tr}(\Sigma_\epsilon)$ for all $0 \leq \alpha \leq 1$. Note that from the standpoint of movement decoding at test time, gesture intention is treated as noise.

The goal is to assess how leveraging the side dataset $(\mathbf{X}', \mathbf{Z}')$ can improve decoding accuracy by understanding the potential characteristics of this total "noise" during testing.

This knowledge helps prevent the decoder from being misled by noise that is aligned with the signal of interest (Y).

Table 4.1 reports the performance of two linear decoders on both training and test data across different regimes of the subspace V (represented by different $0 \leq \alpha \leq 1$ values). In this table, the first decoder, denoted by A_0, is the optimal linear decoder trained solely on the main dataset $(\mathbf{X}, \mathbf{Y})$ and does not utilize the side dataset. The table also includes results for the hypothetical optimal decoder A_{opt}^{test}, which is the optimal linear estimator for the test distribution described above in model (4.14), for comparison.

Moreover, MSE^{train} refers to the relative mean squared error (MSE) of the corresponding decoder applied on training dataset, i.e.,

$$MSE^{train}(A) = \frac{\mathbb{E}[\|Y - \widehat{Y}\|^2]}{\mathrm{Tr}(\mathrm{cov}(Y))} = \frac{\mathrm{Tr}\left(A^\top \Sigma_X A - 2A^\top \Sigma_{XY} + \Sigma_Y\right)}{\mathrm{Tr}(\Sigma_Y)}$$

where A can be any of the two decoders, while MSE^{test} represents the relative MSE on the test model as described in model (4.14), i.e.,

$$MSE^{test}(A) = \frac{\mathbb{E}[\|Y^{test} - \widehat{Y^{test}}\|^2]}{\mathrm{Tr}(\mathrm{cov}(Y^{test}))} = \frac{\mathrm{Tr}\left(A^\top \Sigma_{X^{test}} A - 2A^\top \Sigma_{X^{test}Y^{test}} + \Sigma_{Y^{test}}\right)}{\mathrm{Tr}(\Sigma_{Y^{test}})}$$

here, A be any of the two decoders.

Note that since the subspace U and the distribution of (X, Y) remain constant across all regimes, the Optimal Linear Estimator (OLE) and its $MSE^{train}(A)$ are fixed for all α values. However, varying V through α changes the test data distribution, leading to different $MSE^{test}(A)$ values, even though the used decoder A_0 is fixed. In the last two columns, however, the decoder A_{opt}^{test} also changes, as the test distribution varies with α.

If $MSE^{test}(A_0) \approx MSE^{train}(A_0)$, then no interference has occurred. In other words, the simultaneous intention of a gesture at test time has not affected the performance of the decoder trained solely on the movement dataset. On the other hand, if $MSE^{test}(A_0) >> MSE^{train}(A_0)$, then interference has occurred, meaning the OLE trained on the movement calibration data would not perform well at test time when

a gesture is simultaneously intended. In this case, by examining the performance of the optimal decoder on the test data A_{opt}^{test}, we can determine if there is potential for improvement or if the inherent overlap in decoding subspaces cannot be mitigated by any means.

	A_0		A_{opt}^{test}	
	MSE^{train}	MSE^{test}	MSE^{train}	MSE^{test}
$\alpha = 0$	0.09	1.31	0.39	0.61
$\alpha = 1$	0.09	0.08	0.09	0.07
$\alpha = 0.4$	0.09	0.81	0.22	0.23

Table 4.1: Decoding Performance Across Subspace Regimes

The analysis of Table 4.1 reveals valuable insights:

- Overlapping Subspaces ($\alpha = 0$): When the movement and gesture subspaces overlap, there is inherent confusion between these signals. Even with knowledge of the test distribution, i.e., by employing A_{opt}^{test}, this fundamental limitation cannot be overcome entirely.

- Orthogonal Subspaces ($\alpha = 1$): In the opposite scenario where the subspaces are entirely orthogonal, there is no interference between movement and gesture signals. Therefore, the optimal linear estimator (OLE) using only the main dataset $(\mathbf{X}, \mathbf{Y})$ is sufficient to achieve nearly optimal performance in the test data.

- Intermediate Regimes ($0 < \alpha < 1$): Here, interference between movement and gesture signals occurs. However, if we have access to the test distribution, there is potential for improvement. This study focuses on this regime, leveraging the side dataset $(\mathbf{X}', \mathbf{Z}')$ and our proposed approaches to mitigate this type of interference and enhance decoding performance.

4.4.2 Comparison of decoders for intermediate subspaces

This section investigates how the proposed decoding methods (detailed in the Methods section) can alleviate interference in the case of intermediate subspaces ($0 < \alpha < 1$). Table 4.2 summarizes the performance of different decoders across various BCI use

scenarios and two training examples. Different decoders, training examples, and test scenarios are described respectively in (I), (II), and (III) below.

(I) Decoders

To evaluate the performance of our proposed decoders, we compare them with several alternatives:

- **Optimal Linear Estimator (OLE)**: (A_0, b_0) from equation (4.3) represents the optimal linear decoder learned directly from the calibration dataset $(\mathbf{X}, \mathbf{Y})$. This serves as a baseline for comparison.
- **Tuned Regularized Decoder**: $(A_{\lambda^*}, b_{\lambda^*})$ described in section 4.2.1.1 represents a decoder trained with a hyperparameter selected using our proposed tuning method. This highlights the effectiveness of our hyperparameter selection approach.
- **Zero-Correlated Decoder**: (A_∞, b_∞) from equation (4.12) represents the optimal linear decoder under the constraint of enforcing zero correlation between the estimated movement and the gesture intention. This serves as a reference point to understand the potential benefits of allowing correlated estimates.
- **Hypothetical Optimal Decoder**: $(A_{opt}^{test}, b_{opt}^{test})$ represents the optimal linear decoder for the test distribution, assuming the unrealistic scenario where the test data distribution is perfectly known. This provides a lower bound on the achievable MSE.
- **Minimum Test Prediction Error Regularized Decoder**: $(A_{\lambda_{min}}, b_{\lambda_{min}})$ represents the decoder trained using equation (4.2) with the hyperparameter λ that minimizes the test-time prediction error. This provides a lower bound on the achievable MSE using the proposed regularization.
- **Minimum Test Prediction Error Ridge Regression Decoder**: $(A^{\alpha_{min}}, b^{\alpha_{min}})$ minimizes the expected loss function at test time, where (A^α, b^α) represents

the linear decoder trained with l_2 regularization (Ridge regression), i.e., $(A^\alpha, b^\alpha) = \arg\min_{(A,b)} \mathbb{E}[L(Y, (XA + b))] + \alpha\|A\|_F^2$. This investigates a lower bound for achievable MSE of the regularization technique employed by Wodlinger et al. and Collinger et al. [28, 29]. Additionally, it helps distinguish the benefits gained from regularization in general from those gained specifically by our approach.

It is important to note that the first three estimators above can be estimated solely from the calibration datasets $(\mathbf{X}, \mathbf{Y})$ and $(\mathbf{X}', \mathbf{Z}')$ (The first one is estimated only by $(\mathbf{X}, \mathbf{Y})$). On the other hand, the remaining decoders require knowledge of the actual distribution of the test data. As in practical settings, this information is often unavailable, these latter decoders are included for comparative purposes.

(II) Training examples

We evaluate the performance of the decoders under two different calibration settings. For both examples, $d_X = 300$, $d_Y = 2$, and $d_Z = 4$.

(A) **Illustrative example: Data-rich environment:** To begin, we introduce an illustrative example that highlights the key principles of our approaches. This example assumes a data-rich environment, where the joint distribution is known for both datasets $(\mathbf{X}, \mathbf{Y})$ and $(\mathbf{X}', \mathbf{Z}')$, i.e., the mean and covariance matrices required for determining the decoders A_λ ($0 \leq \lambda \leq \infty$) in equations (4.3) and (4.12) are known. This simplified environment allows for a clear demonstration of the relationships between different decoders and their properties. The example used in the previous section (4.4.1) for $\alpha = 0.4$ employs the same underlying training parameters as in example (A) here.

(B) **Realistic Scenario: Inference from Calibration Data:** In a more realistic scenario, unlike the data-rich case, we do not have full knowledge of the underlying neural encoding models. Instead, we start with simulated synthetic calibration datasets $(\mathbf{X}, \mathbf{Y})$ and $(\mathbf{X}', \mathbf{Z}')$. From these datasets, we estimate the covariance matrices necessary for computing the decoders A_λ

(for $0 \leq \lambda \leq \infty$) and then apply these estimates in the analytic formulas to obtain the solutions.

In this example, after generating each entry of U from a standard normal distribution and normalizing each row to have unit length, we use an intermediate regime ($\alpha = 0.4$) to construct V. We then simulate $\mathbf{Y}$ and $\mathbf{Z}'$ from $N_{d_Y}(0, I)$ and $N_{d_Z}(0, 1.5I)$ with $n = n' = 10^5$, respectively. After simulating $\mathbf{Y}$ and $\mathbf{Z}'$, we generate $\mathbf{X}$ and $\mathbf{X}'$ using the encoding models in (4.7), with $\Sigma_\epsilon = 0.3I$ and $\Sigma_{\epsilon'} = 0.2I$.

(III) BCI use scenarios

Now for BCI use, the generic model we employ is $X^{test} = Y^{test}U + Z^{test}V + \epsilon^{test}$, where noise term ϵ^{test} is independent from Y^{test} and Z^{test}, and it is drawn from a multivariate Gaussian $N_{d_X}(0, \Sigma_{\epsilon^{test}})$. Y^{test} and Z^{test} are also assumed independent in this model.

We assess the performance of different decoders in the following different test scenarios:

(a) **Movement-only**: This scenario represents a BCI use case where the user solely intends to perform a specific movement task, identical to the task practiced during movement calibration. We model this as having joint distribution of X, Y, and Z during BCI use coincides with the joint distribution observed during movement calibration, i.e., $\Sigma_{Y^{test}} = \Sigma_Y$, $\Sigma_{\epsilon_{test}} = \Sigma_\epsilon$, and $Z^{test} = 0$ (as the user has exhibited no intentional gesture alongside the movement during movement calibration).

(b) **Concurrent movement and gesture**: This scenario reflects the situation where the user intends to perform a movement while simultaneously engaging in a specific gesture. This situation directly addresses the core question of this study: how can we improve each decoder's performance given that calibration data might only capture movement or gesture individually. In this

case, $\Sigma_{Y^{test}} = \Sigma_Y$, $\Sigma_{\epsilon_{test}} = \frac{1}{2}(\Sigma_\epsilon + \Sigma_{\epsilon'})$, and $\Sigma_{Z^{test}} = \Sigma_{Z'}$.

(c) **Enhanced gesture influence**: This scenario builds upon the previous one by introducing an additional complication. Here, the user's gesture intention exerts a stronger influence on the neural signal during BCI use compared to what was observed in the gesture calibration data. We model this by $\Sigma_{Y^{test}} = \Sigma_Y$, $\Sigma_{\epsilon_{test}} = \frac{1}{2}(\Sigma_\epsilon + \Sigma_{\epsilon'})$, and $\Sigma_{Z^{test}} = s_Z\Sigma_{Z'}$ for varying the scaling factor $s_Z \in \{1, 2, \cdots, 10\}$. This case will be explored in section 4.4.3.

For data-rich training (example (A)), the covariance matrices (Σ_Y, $\Sigma_{Z'}$, Σ_ϵ, $\Sigma_{\epsilon'}$) and subspace matrices U and V used in the test models above correspond to the true underlying parameters. In the more realistic scenario (example (B)), these parameters are estimates obtained from the calibration datasets.

Example	BCI use	A_0	A_{λ^*}	A_∞	A_{opt}^{test}	$A_{\lambda_{min}}$	$A^{\alpha_{min}}$
(A)	(a)	0.09	0.22	0.33	0.09	0.09	0.09
	(b)	0.81	0.24	0.29	0.23	0.24	0.49
(B)	(a)	0.23	0.34	0.60	0.23	0.23	0.23
	(b)	0.76	0.41	0.56	0.40	0.41	0.54

Table 4.2: Comparison between predictive performance of different decoders.

The analysis of Table 4.2 reveals several key findings:

- **Performance under no distribution shift** For both the data-rich example and the more realistic high-dimensional scenario, performance degrades when there is no intention of gesture (no distribution shift). This is expected, since in case of no distribution shift, the optimal decoder at test time is the same as the OLE derived from the movement training dataset, therefore, any regularization with $\lambda \neq 0$ would introduce unnecessary penalty and results in performance degradation.

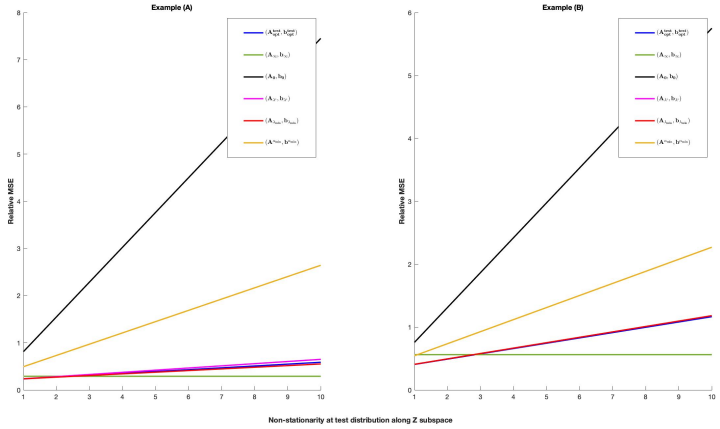
- **Performance under interference** However, when the user intends both movement and gesture during BCI use (distribution shift), our proposed methods demonstrate a substantial advantage. By leveraging information from the gesture calibration

dataset $(\mathbf{X}', \mathbf{Z}')$, both the tuned regularization decoder $(A_{\lambda^*}, b_{\lambda^*})$ and the uncorrelated approach (A_∞, b_∞) achieve significantly better performance compared to relying solely on the OLE derived from the movement calibration data. Notably, regularization can nearly approach the level of optimal performance in this scenario, as evidenced by the performance of the decoder associated with $(A_{\lambda_{min}}, b_{\lambda_{min}})$, which closely matches the optimal decoder performance in the column $(A_{opt}^{test}, b_{opt}^{test})$ (known only for evaluation purposes).

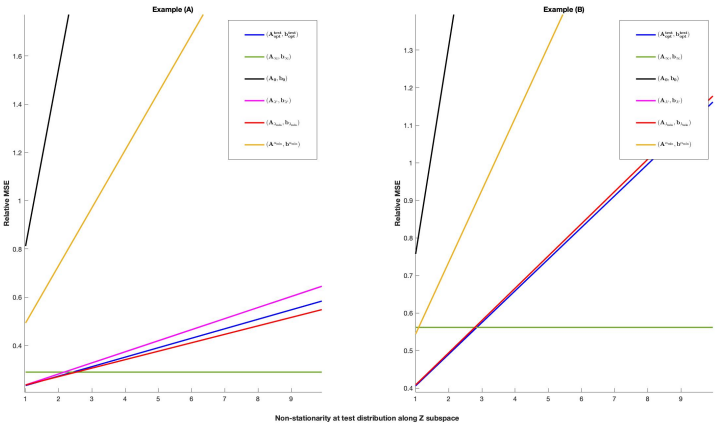
- **Comparison with l_2 Regularization** The final column of Table 4.2 displays the lowest achievable MSE using l_2 regularization (see Section (I) for details). Our results demonstrate that in the scenario where the user intends both movement and gesture (distribution shift), both of our proposed methods outperform l_2 regularization significantly.

4.4.3 Robustness of the uncorrelated decoder

Figure 4.1 demonstrates the relative MSE of different decoders in the BCI test scenario (c) for both examples (A) and (B). The test distribution is varied by the scaling factor s_Z, which enhances the influence of gestures at test time. While the hypothetical optimal decoder $(A_{opt}^{test}, b_{opt}^{test})$ performs slightly better when the user's gestures have minimal influence (low values of s_Z in the scenario described in Section (c)), its performance degrades rapidly as the user's gestures become more influential during BCI use (represented by increasing values of s_Z). In contrast, the uncorrelated decoder (A_∞, b_∞) proposed in this study exhibits full robustness, maintaining its predictive power even under this distribution shift caused by stronger gesture influence. Notably, the best achievable decoder using l_2 regularization, $(A^{\alpha_{min}}, b^{\alpha_{min}})$, also shows limited robustness.



(a) Relative MSE of decoders versus relative HSIC, for varying scaling factors s_Z and s_δ



(b) Figure 4.1a zoomed in.

Figure 4.1: Comparison between different decoders in both examples, under enhanced gesture influence scenario (varying scaling factor s_Z from 1 to 10)

4.5 Conclusion

In multi-task BCI systems, performance degradation often occurs when participants engage in two simultaneous MI tasks. To address this challenge, we proposed two approaches in this paper. The first approach involved incorporating a new calibration method that includes a regularization term designed to penalize the statistical dependence between the decoded primary task and the interfering secondary intention. The second approach involved transforming the neural recording data into a subspace where any linear decoder remains completely uncorrelated with the interfering task. This transformation offers the added advantage of not requiring a new calibration session, as the

transformed neural data can be fed directly into any pre-existing decoder during the pre-processing step. Both methods demonstrated improved performance in synthetic data when both tasks were simultaneously intended during testing, compared to using a decoder trained solely on the primary task's dataset.

Our methods offer a significant advantage: the interfering source, represented by Z, can be any quantifiable noise source that affects the primary task decoder, as long as a surrogate dataset is available. This flexibility means that Z is not limited to another task intention, but can encompass various types of noise. Furthermore, we can show that in the loss function of the first method, the Ordinary Least Squares Estimator (OLE) of Z can be replaced by Z itself. This implies that if true labels for Z are unavailable, any accurate decoder for Z can be substituted, expanding the applicability of our approach.

A more suitable approach could involve ensuring that the decoded primary task $\widehat{Y}$ and the intended secondary task are independent, a concept known as demographic (or statistical) parity in the field of fairness in machine learning [88, 94]. Chzhen et al. demonstrated that achieving this independence is feasible when the variable Z is available at prediction time [94]. However, a unique challenge in this study, as opposed to the typical scenarios in the field of fairness in machine learning, is that the secondary variable Z is not known at prediction time.

An alternative and potentially more effective penalty term for the first approach could involve using the coefficient of determination (an extension of the correlation coefficient) instead of covariance in (4.1), as explored by [112]. However, their non-convex optimization algorithm also assumes that Z is available during prediction. A limitation of our method is that the penalty term can be eliminated by converging A to zero, an issue that would not arise if the coefficient of determination were used. To address this limitation, one possible solution could be to constrain A in (4.2) such that $\|A\|^2 = c$ for a constant c, which could be a direction for future research.

4.6 Supplementary material

4.6.1 Derivation of equation (4.3)

First, assume that $\mathbb{E}[X] = \mathbb{E}[Y] = \mathbb{E}[X'] = \mathbb{E}[Z'] = \mathbf{0}$.

Let

$$A_\lambda^0 \triangleq \arg\min_A \mathbb{E}[\|XA - Y\|^2] + \lambda \|\text{cov}(X'A, Z')\|_F^2 \tag{4.15}$$

Note that

$$\begin{aligned}
\mathbb{E}[\|XA - Y\|^2] &= \mathbb{E}\left[\text{Tr}\left((XA - Y)^\top (XA - Y)\right)\right] \\
&= \mathbb{E}\left[\text{Tr}\left(A^\top X^\top XA\right) - \text{Tr}(A^\top X^\top Y) - \text{Tr}(Y^\top XA) + \text{Tr}(Y^\top Y)\right] \\
&= \mathbb{E}\left[\text{Tr}\left(A^\top X^\top XA\right)\right] - \mathbb{E}\left[2\,\text{Tr}(A^\top X^\top Y)\right] + \mathbb{E}\left[\text{Tr}(Y^\top Y)\right] \\
&= \text{Tr}\left(\mathbb{E}\left[A^\top X^\top XA\right] - 2\,\mathbb{E}[A^\top X^\top Y] + \mathbb{E}[Y^\top Y]\right) \\
&= \text{Tr}\left(A^\top \Sigma_X A - 2A^\top \Sigma_{XY} + \Sigma_Y\right)
\end{aligned} \tag{4.16}$$

And,

$$\begin{aligned}
\|cov\,(X'A, Z')\,\|_F^2 &= \|\,\mathbb{E}[Z'^\top X'A]\|_F^2 \\
&= \|\,\mathbb{E}[Z'^\top X']A\|_F^2 \\
&= \|\Sigma_{Z'X'}A\|_F^2 \\
&= \text{Tr}\left(A^\top \Sigma_{Z'X'}^\top \Sigma_{Z'X'} A\right)
\end{aligned} \tag{4.17}$$

Therefore,

$$\begin{aligned}
A_\lambda^0 &= \arg\min_A \left(\mathbb{E}[\|XA - Y\|^2] + \lambda \|\text{cov}\,(X'A, Z')\,\|_F^2\right) \\
&= \arg\min_A H_\lambda(A)
\end{aligned} \tag{4.18}$$

where $H_\lambda(A) = \text{Tr}\left(A^\top \left(\Sigma_X + \lambda \Sigma_{Z'X'}^\top \Sigma_{Z'X'}\right) A - 2A^\top \Sigma_{XY}\right)$, and we have (see [113], [114]):

$$\frac{\partial}{\partial A} H_\lambda(A) = \left(\left(\Sigma_X + \lambda \Sigma_{Z'X'}^\top \Sigma_{Z'X'} \right) + \left(\Sigma_X + \lambda \Sigma_{Z'X'}^\top \Sigma_{Z'X'} \right)^\top \right) A - 2\Sigma_{XY} \quad (4.19)$$

Σ_X is assumed to be positive definite, therefore, since $\Sigma_{X'Z'}\Sigma_{Z'X'}$ is positive-semidefinite, the Hessian matrix $\frac{\partial^2}{\partial A \partial A^\top} H_\lambda(A) = \left(\Sigma_X + \lambda \Sigma_{Z'X'}^\top \Sigma_{Z'X'} \right)$ is symmetric positive-definite for $\lambda > 0$, and we have

$$A_\lambda^0 = \left(\Sigma_X + \lambda \Sigma_{X'Z'} \Sigma_{Z'X'} \right)^{-1} \Sigma_{XY} \quad (4.20)$$

Now in a general case where $\mathbb{E}[X] = \mu_X$, $\mathbb{E}[Y] = \mu_Y$, $\mathbb{E}[X'] = \mu_{X'}$ and $\mathbb{E}[Z'] = \mu_{Z'}$, note that from the definition of A_λ and b_λ in (4.2) in the main text, we have

$$\begin{aligned} (A_\lambda, b_\lambda) &= \arg\min_{(A,b)} \mathbb{E}[\|(XA + b) - Y\|^2] + \lambda \|\text{cov}(X'A + b, Z')\|_F^2 \\ &= \arg\min_{(A,b)} \mathbb{E}[\|(XA + b) - Y\|^2] + \lambda \|\text{cov}(X'A, Z')\|_F^2 \end{aligned} \quad (4.21)$$

Now if we define $b_A^* = \arg\min_b \left(\mathbb{E}[\|(XA + b) - Y\|^2] + \lambda \|\text{cov}(X'A, Z')\|_F^2 \right)$, then

$$A_\lambda = \arg\min_A \left(\mathbb{E}[\|(XA + b_A^*) - Y\|^2] + \lambda \|\text{cov}(X'A, Z')\|_F^2 \right) \quad (4.22)$$

But we have

$$\begin{aligned} b_A^* &= \arg\min_b \left(\mathbb{E}[\|(XA + b) - Y\|^2] + \lambda \|\text{cov}(X'A, Z')\|_F^2 \right) \\ &= \arg\min_b \mathbb{E}[\|(XA + b) - Y\|^2] \\ &= \mathbb{E}[Y - XA] \\ &= \mu_Y - \mu_X A \end{aligned} \quad (4.23)$$

Therefore, in (4.22), we have

$$\begin{aligned} A_\lambda &= \arg\min_A \left(\mathbb{E}[\|(XA + \mu_Y - \mu_X A) - Y\|^2] + \lambda \|\text{cov}(X'A, Z')\|_F^2\right) \\ &= \arg\min_A (\mathbb{E}[\|(X - \mu_X)A - (Y - \mu_Y)\|^2] \\ &\qquad + \lambda \|\text{cov}\left((X' - \mu_{X'})A, Z' - \mu_{Z'}\right)\|_F^2) \end{aligned} \tag{4.24}$$

which means that we are back in the case of (4.15). Therefore, $A_\lambda = A_\lambda^0$, and this completes the proof.

4.6.2 Proofs of section 4.2.2

4.6.2.1 Proof of lemma 11

Lemma 11: Let $\Sigma_{Z'X'} \triangleq \text{cov}(Z', X')$ be well-defined and finite-valued. The following are equivalent properties of a matrix H (necessarily with d_X rows):

(a) A satisfies $\|\text{cov}(X'A, Z')\|_F^2 = 0$ if and only if $A = HB$ for some B.

(b) The column space of H is equal to the null space of $\Sigma_{Z'X'}$.

If F is a $d_X \times d_X$ symmetric positive-definite matrix and $\Sigma_{Z'X'} F \Sigma_{X'Z'}$ is invertible, then

$$H = F - F\Sigma_{X'Z'}(\Sigma_{Z'X'} F \Sigma_{X'Z'})^{-1}\Sigma_{Z'X'} F$$

satisfies (a) and (b).

Proof. **Note**: Let A_i denote the i-th column of A, then from (4.17):

$$\begin{aligned} \|\text{cov}(X'A, Z')\|_F^2 &= \text{Tr}\left(A^\top \Sigma_{Z'X'}^\top \Sigma_{Z'X'} A\right) \\ &= \sum_{i=1}^{d_Y} A_i^\top \Sigma_{Z'X'}^\top \Sigma_{Z'X'} A_i \\ &= \sum_{i=1}^{d_Y} \|\Sigma_{Z'X'} A_i\|^2 \end{aligned} \tag{4.25}$$

Therefore, $\|\text{cov}(X'A, Z')\|_F^2 = 0$ is equivalent to $\Sigma_{Z'X'}A_i = 0$ for all $i \in \{1, \cdots, d_Y\}$.

First:

(b) to (a): Let H be any matrix with column space equal to null$(\Sigma_{Z'X'})$. Then if $A = HB$ for some B, then each column of A will lie in null$(\Sigma_{Z'X'})$, hence $\|\text{cov}(X'A, Z')\|_F^2 = 0$ from the note above. Conversely, if $\|\text{cov}(X'A, Z')\|_F^2 = 0$, then again from the note, each column of A must lie in the null space of the covariance matrix $\Sigma_{Z'X'}$, and as the column space of H is null$(\Sigma_{Z'X'})$, there exists some B such that $A = HB$.

(a) to (b): Let H satisfies the property in (a). We first show that for every $v \in$ null$(\Sigma_{Z'X'})$, v is in the column space of H. Let $A_i = v$ for every i. Then from the note, $\|\text{cov}(X'A, Z')\|_F^2 = 0$, therefore, $A = HB$ for some B, which means that v is in the column space of H. On the other hand, for any $u \in \mathbb{R}^{d_X}$ that is in the column space of H, the matrix A such that $A_i = u$ for all i can be written as HB for some B, therefore, by the property in (a) and the note, $u \in$ null$(\Sigma_{Z'X'})$.

Second:

First note that for $H = F - F\Sigma_{X'Z'}(\Sigma_{Z'X'}F\Sigma_{X'Z'})^{-1}\Sigma_{Z'X'}F$, if $A = HB$, then $\Sigma_{Z'X'}A = 0$ because $\Sigma_{Z'X'}H = 0$, so one direction is trivial.

For the other direction, we need to show that $\|\text{cov}(X'A, Z')\|_F^2 = 0$ only if $A = HB$ for some B. let us first assume that $F = I$:

If $\|\text{cov}(X'A, Z')\|_F^2 = 0$, or equivalently (from the note), $\Sigma_{Z'X'}A = 0$, then A can be written as $A = (I - \Sigma_{X'Z'}(\Sigma_{Z'X'}\Sigma_{X'Z'})^{-1}\Sigma_{Z'X'})A$. This completes the proof for $F = I$.

Now if F is any symmetric positive-definite matrix, let us define $\tilde{X} \triangleq X'F^{1/2}$. Then $\Sigma_{Z'\tilde{X}} = \Sigma_{Z'X'}F^{1/2}$. From the already completed part of proof for $F = I$, we know that the matrix below:

$$\tilde{H} = I - \Sigma_{\tilde{X}Z'}(\Sigma_{Z'\tilde{X}}\Sigma_{\tilde{X}Z'})^{-1}\Sigma_{Z'\tilde{X}} = I - F^{1/2}\Sigma_{X'Z'}(\Sigma_{Z'X'}F\Sigma_{X'Z'})^{-1}\Sigma_{Z'X'}F^{1/2}$$

has the property that $\|\text{cov}(\tilde{X}\tilde{A}, Z')\|_F^2 = 0$ if and only if $\tilde{A} = \tilde{H}\tilde{B}$ for some $\tilde{B}$. Now if $\|\text{cov}(X'A, Z')\|_F^2 = 0$, then $\|\text{cov}(\tilde{X}F^{-1/2}A, Z')\|_F^2 = 0$, which implies that $F^{-1/2}A = \tilde{H}\tilde{B}$ for some $\tilde{B}$, thus

$$A = F^{1/2}\tilde{H}F^{1/2}(F^{-1/2}\tilde{B}) = (F - F\Sigma_{X'Z'}(\Sigma_{Z'X'}F\Sigma_{X'Z'})^{-1}\Sigma_{Z'X'}F)B$$

for $B = F^{-1/2}\tilde{B}$, and this completes the proof.

□

4.6.2.2 Proof of Theorem 12

The optimization problem (4.11) in the main text was as below:

$$\underset{\substack{(A,b) \\ \|\text{cov}(X'A,Z')\|_F^2=0}}{\arg\min} \quad \mathbb{E}[\|Y - (XA + b)\|^2]$$

Theorem 12: Let Σ_X, Σ_Y, $\Sigma_{X'}$, and $\Sigma_{Z'}$ be well-defined and finite-valued.

(a) Let $H \in \mathbb{R}^{d_X \times t}$ be any matrix with column space equal to null$(\Sigma_{Z'X'})$. If $(B^*, c^*) \in \mathbb{R}^{t \times d_Y} \times \mathbb{R}^{1 \times d_Y}$ minimizes $\mathbb{E}[\|Y - (XHB + c)\|^2]$ over (B, c), then $(A^*, b^*) = (HB^*, c^*)$ is a solution to (4.11).

(b) If $A_\infty = \lim_{\lambda \to \infty} A_\lambda$ and $b_\infty = \lim_{\lambda \to \infty} b_\lambda$ exist, then (A_∞, b_∞) is a solution to (4.11).

(c) If $\Sigma_{Z'X'}$ and Σ_X are both of full rank, then (4.11) has the following unique closed-form solution

$$A_\infty = \Sigma_X^{-1}\left(I - \Sigma_{X'Z'}(\Sigma_{Z'X'}\Sigma_X^{-1}\Sigma_{X'Z'})^{-1}\Sigma_{Z'X'}\Sigma_X^{-1}\right)\Sigma_{XY}$$
$$b_\infty = \mu_Y - \mu_X A_\infty$$

Proof. (a) Let $G = \{A \in \mathbb{R}^{d_X \times d_Y} : \|\text{cov}(X'A, Z')\|_F^2 = 0\}$. Then by lemma 11, $G = \{HB : B \in \mathbb{R}^{t \times d_Y}\}$. So for each (A, b) such that $A \in G$, we have $A = HB$ for some B, thus by the definition of (B^*, c^*) we have

$$\mathbb{E}[\|Y-(XHB^*+c^*)\|^2] \leq \mathbb{E}[\|Y-(XHB+b)\|^2] = \mathbb{E}[\|Y-(XA+b)\|^2]$$

which shows that $(A^*, b^*) = (HB^*, c^*)$ minimizes $\mathbb{E}[\|Y-(XA+b)\|^2]$ over (A, b) such that $A \in G$.

(b) Now assume that $A_\infty = \lim_{\lambda\to\infty} A_\lambda$ exists. Note that then, from (4.3) in the main text, we have $b_\infty = \mu_Y - \mu_X A_\infty$ also exists. We want to show that (A_∞, b_∞) minimizes $\mathbb{E}[\|Y-(XA+b)\|^2]$ over (A, b) such that $A \in G$.

Let $f(A, b) = \mathbb{E}[\|Y-(XA+b)\|^2]$, and $g(A) = \|\text{cov}(X'A, Z')\|_F^2$. Then

$$(A_\lambda, b_\lambda) = \arg\min_{(A,b)} f(A, b) + \lambda g(A)$$

for $0 \leq \lambda < \infty$.

First we show that $g(A_\infty) = 0$, or equivalently, $A_\infty \in G$. Note that f and g are both non-negative and continuous functions. Let S be any element in the non-empty set G. For the sake of contradiction, we assume that $g(A_\infty) = \lim_{\lambda\to\infty} g(A_\lambda) > 0$. Then there exists $\epsilon > 0$ such that $\lim_{\lambda\to\infty} g(A_\lambda) > \epsilon$. Thus there exists $M > 1 + \frac{f(S,0)-f(A_\infty,b_\infty)}{\epsilon}$ such that $\forall\lambda > M$, we have $g(A_\lambda) > \epsilon$ and $f(A_\lambda, b_\lambda) > f(A_\infty, b_\infty) - \epsilon$ (since $\lim_{\lambda\to\infty} f(A_\lambda, b_\lambda) = f(A_\infty, b_\infty)$). Therefore, $\forall\lambda > M$ we have

$$f(A_\lambda, b_\lambda) + \lambda g(A_\lambda) > f(A_\infty, b_\infty) - \epsilon + (1 + \frac{f(S,0) - f(A_\infty, b_\infty)}{\epsilon})\epsilon = f(S,0) + g(S)$$

which contradicts with the definition of (A_λ, b_λ).

Next, we note that for any $(\tilde{A}, \tilde{b})$ which minimizes $\|\text{cov}(X'A, Z')\|_F^2 = 0$ over (A, b) such that $A \in G$, by definition of A_λ we have

$$f(\tilde{A}, \tilde{b}) = f(\tilde{A}, \tilde{b}) + \lambda g(\tilde{A}) \geq f(A_\lambda, b_\lambda) + \lambda g(A_\lambda)$$

for all $\lambda \geq 0$.

Therefore, $f(\tilde{A},\tilde{b}) - f(A_\lambda, b_\lambda) \geq \lambda g(A_\lambda)$ for all $\lambda \geq 0$, thus by taking limit of both sides, we get $f(\tilde{A},\tilde{b}) - f(A_\infty, b_\infty) \geq \lim_{\lambda\to\infty} \lambda g(A_\lambda)$, and since $\lim_{\lambda\to\infty} \lambda g(A_\lambda)$ is non-negative, we conclude that $f(\tilde{A},\tilde{b}) \geq f(A_\infty, b_\infty)$. We previously showed that $A_\infty \in G$, which implies that $f(\tilde{A},\tilde{b}) \leq f(A_\infty, b_\infty)$. Therefore, $f(\tilde{A},\tilde{b}) = f(A_\infty, b_\infty)$. So the proof is complete.

(c) For simplicity , let us assume that $\mathbb{E}[X] = \mathbb{E}[Y] = \mathbb{E}[X'] = \mathbb{E}[Z'] = \mathbf{0}$. Extending this derivation to the case of X and Y with non-zero mean is the same procedure as in section 4.6.1, i.e., equations (4.21), (4.22), (4.23) and (4.24). The optimization problem in (4.11) in the main text can be written as below:

$$\underset{\substack{A \\ \Sigma_{Z'X'}A_i=0 \ \forall i}}{\arg\min} \ \mathbb{E}[\|Y - XA\|^2] = \underset{\substack{A \\ \Sigma_{Z'X'}A=0}}{\arg\min} \ \mathbb{E}[\|Y - XA\|^2] \tag{4.26}$$

Therefore, it is a constrained least squares (CLS) problem described in [115], whose solution (since Σ_X and $\Sigma_{Z'X'}$ are assumed to have full rank in this case) is as below:

$$\begin{aligned} A_\infty &= A_{opt} - \Sigma_X^{-1}\Sigma_{Z'X'}^\top(\Sigma_{Z'X'}\Sigma_X^{-1}\Sigma_{Z'X'}^\top)^{-1}\Sigma_{Z'X'}A_{opt} \\ &= \Sigma_X^{-1}\Sigma_{XY} - \Sigma_X^{-1}\Sigma_{Z'X'}^\top(\Sigma_{Z'X'}\Sigma_X^{-1}\Sigma_{Z'X'}^\top)^{-1}\Sigma_{Z'X'}\Sigma_X^{-1}\Sigma_{XY} \\ &= \Sigma_X^{-1}\left(I - \Sigma_{X'Z'}(\Sigma_{Z'X'}\Sigma_X^{-1}\Sigma_{X'Z'})^{-1}\Sigma_{Z'X'}\Sigma_X^{-1}\right)\Sigma_{XY} \end{aligned} \tag{4.27}$$

Note that A_{opt} is the optimal linear estimator (OLE) of unconstrained least squares, which is $\Sigma_X^{-1}\Sigma_{XY}$ in this case that Σ_X is nonsingular (See (4.3)). Therefore, the constrained optimization problem in (4.11) in the main text has a unique solution in this case. □

Corollary 13. *If Σ_X is invertible, then* (4.12) *(in the main text) is a solution to the optiomization problem* (4.11) *(in the main text).*

Proof. We show that if Σ_X is invertible, then $\lim_{\lambda\to\infty} A_\lambda$ and $\lim_{\lambda\to\infty} b_\lambda$ exist and is equal to (4.12), and therefore by theorem 12 (part (b)), it is a solution to the optimization problem (4.11) in the main text.

$\lim_{\lambda\to\infty} A_\lambda$ can be found using the Woodbury matrix identity for invertible matrices A, B, and $B^{-1} + C^\top A^{-1} C$, which is as follows [116]:

$$(A + CBC^\top)^{-1} = A^{-1} - A^{-1}C(B^{-1} + C^\top A^{-1}C)^{-1}C^\top A^{-1} \tag{4.28}$$

Let $A = \Sigma_X$, $B = \lambda I$ for $\lambda > 0$ and $C = \Sigma_{X'Z'}$. Then we have A invertible because Σ_X is assumed invertible in this case, and we also have $B^{-1} + C^\top A^{-1} C$ invertible, because $\Sigma_{Z'X'}\Sigma_X^{-1}\Sigma_{X'Z'}$ is positive semidefinite, and $B^{-1} = \frac{1}{\lambda} I$ is positive definite, thus $B^{-1} + C^\top A^{-1} C$ is positive definite. Therefore, using this identity to rewrite A_λ in (4.20) we have

$$A_\lambda = \left(\Sigma_X^{-1} - \Sigma_X^{-1}\Sigma_{X'Z'}(\frac{1}{\lambda} I + \Sigma_{Z'X'}\Sigma_X^{-1}\Sigma_{X'Z'})^{-1}\Sigma_{Z'X'}\Sigma_X^{-1}\right)\Sigma_{XY} \tag{4.29}$$

Now since the matrix inverse is a continuous map, limit and inverse can interchange, hence we have

$$\begin{aligned}\lim_{\lambda\to\infty} A_\lambda &= \lim_{\lambda\to\infty}\left(\Sigma_X^{-1} - \Sigma_X^{-1}\Sigma_{X'Z'}(\frac{1}{\lambda} I + \Sigma_{Z'X'}\Sigma_X^{-1}\Sigma_{X'Z'})^{-1}\Sigma_{Z'X'}\Sigma_X^{-1}\right)\Sigma_{XY} \\ &= \left(\Sigma_X^{-1} - \Sigma_X^{-1}\Sigma_{X'Z'}(\Sigma_{Z'X'}\Sigma_X^{-1}\Sigma_{X'Z'})^{-1}\Sigma_{Z'X'}\Sigma_X^{-1}\right)\Sigma_{XY}\end{aligned} \tag{4.30}$$

which is (4.12) in the main text.

□

Corollary 14. *If $\Sigma_{Z'X'}$ is of full rank, and there exists a matrix $H \in \mathbb{R}^{d_X \times (d_X - d_Z)}$ with column space equal to* null$(\Sigma_{Z'X'})$*, such that $H^\top \Sigma_X H$ is non-singular, then the constrained optimization problem* (4.11) *has a solution below:*

$$\begin{aligned} A_\infty &= H(H^\top \Sigma_X H)^{-1} H^\top \Sigma_{XY} \\ b_\infty &= \mu_Y - \mu_X A_\infty \end{aligned} \tag{4.31}$$

which is not necessarily unique, unless Σ_X is non-singular.

Proof. In part (a) of the proof of theorem 12, we showed that if B^* is the optimal linear estimator (OLE) of Y given $T(X) = XH$, then HB^* minimizes the optimization problem in (4.11). Since $H^\top \Sigma_X H$ is non-singular, B^* has the unique form of $B^* = \Sigma_{(XH)}^{-1} \Sigma_{XH,Y}$, so we have

$$\begin{aligned} A^* &= HB^* = H\Sigma_{(XH)}^{-1}\Sigma_{XH,Y} = H(H^\top \Sigma_X H)^{-1} H^\top \Sigma_{XY} \\ b^* &= \mu_Y - \mu_X A^* \end{aligned} \tag{4.32}$$

minimizes $\mathbb{E}[\|Y - T(X)A + b\|^2]$ over (A, b) such that $\|\text{cov}(X'A, Z')\|_F^2 = 0$.

This result could also be drawn from [115], where the solution of this CLS with constraint $\|\text{cov}(X'A, Z')\|_F^2 = 0$, or equivalently $\Sigma_{Z'X'}A = 0$ is shown to have the form

$$R(R^\top \Sigma_X R)^{-1} R^\top \Sigma_{XY}$$

given that $R^\top \Sigma_X R$ is invertible. Here, $R \in \mathbb{R}^{d_X \times (d_X - d_Z)}$ is the matrix such that $[\Sigma_{Z'X'}^\top \; R]$ is nonsingular, and $R^\top \Sigma_{Z'X'}^\top = 0$. Note that any R that satisfies these conditions is a matrix $H \in \mathbb{R}^{d_X \times (d_X - d_Z)}$ with column space equal to null$(\Sigma_{Z'X'})$, with $H^\top \Sigma_X H$ invertible. Therefore, it is the same result.

□

Remark 7. In case (c) of Theorem 12, which is the case of full rank assumption for Σ_X and $\Sigma_{Z'X'}$, it is worth it to show that that the unique solution found in (4.27) is the same as (4.32) for a $H \in \mathbb{R}^{d_X \times (d_X - d_Z)}$ with column space equal to null$(\Sigma_{Z'X'})$. This can be the second proof for part (c), using the corollary 14 that only uses part (a).

We want to show the following equation

$$H(H^\top \Sigma_X H)^{-1} H^\top = \Sigma_X^{-1} - \Sigma_X^{-1}\Sigma_{X'Z'}(\Sigma_{Z'X'}\Sigma_X^{-1}\Sigma_{X'Z'})^{-1}\Sigma_{Z'X'}\Sigma_X^{-1} \tag{4.33}$$

As Σ_X is positive-definite, by multiplying both sides of (4.33) by $\Sigma_X^{1/2}$ on the left and right, we have

$$U(U^\top U)^{-1}U^\intercal = I - V(V^\top V)^{-1}V^\intercal \tag{4.34}$$

where $U = \Sigma_X^{1/2}H$ and $V = \Sigma_X^{-1/2}\Sigma_{X'Z'}$. Note that since Σ_X is invertible, so are $H^\top\Sigma_X H$ and $\Sigma_{Z'X'}\Sigma_X^{-1}\Sigma_{X'Z'}$, as $H \in \mathbb{R}^{d_X\times(d_X-d_Z)}$ and $\Sigma_{Z'X'}$ are both of full rank.

Now note that the left side of (4.34) is the orthogonal projection matrix onto the column space of U, and the right hand side is the orthogonal projection matrix onto the $W^\perp$ where W is the column space of V, or equivalently, the null space of $V^\top$. Therefore, it remains to show that the the null space of $V^\top$ is equal to the column space of U:

$$V^\top U = \Sigma_{Z'X'}\Sigma_X^{-1/2}\Sigma_X^{1/2}H = \Sigma_{Z'X'}H = 0$$

On the other hand, if $V^\top c = 0$ for $c \in \mathbb{R}^{d_X}$, then $\Sigma_{Z'X'}\Sigma_X^{-1/2}c = 0$, therefore, $\Sigma_X^{-1/2}c$ is in the null space of $\Sigma_{Z'X'}$, thus in the column space of H, i.e., $\Sigma_X^{-1/2}c = Hz$ for some z, hence $c = \Sigma_X^{1/2}Hz = Uz$, which means that c is in the column space of U, which completes the proof.

4.6.3 Relative loss for decoder training

Training the decoder A_λ may be done using this normalized objective function as below:

$$A_\lambda^{rel} = \arg\min_A \left(\frac{\mathbb{E}[\|XA - Y\|^2]}{\mathrm{Tr}(\mathrm{cov}(Y))} + \lambda \frac{\|\mathrm{cov}\,(X'A, Z')\,\|_F^2}{\left(\frac{d_Y}{d_X}\right)\mathrm{Tr}(\mathrm{cov}(X'))\,\mathrm{Tr}(\mathrm{cov}(Z'))} \right) \tag{4.35}$$

This choice offers invariance with respect to unit scaling and data repetition (will be explored below). In fact we have

$$A_\lambda^{rel} = A_{t\lambda} \quad \text{where} \quad t = \frac{\mathrm{Tr}(\mathrm{cov}(Y))}{\left(\frac{d_Y}{d_X}\right)\mathrm{Tr}(\mathrm{cov}(X'))\,\mathrm{Tr}(\mathrm{cov}(Z'))}$$

where A_λ is defined in (4.3) in the main text. We also have $A_\infty^{rel} = A_\infty$.

For any result shown in this study, when referring to A_λ, we are specifically referring to A_λ^{rel}. Furthermore, all mentions of MSE will be in their relative form.

4.6.3.1 Invariance with respect to unit scaling and orthogonal change of basis for Z'

We first show that for each $\lambda \geq 0$:

$$A_\lambda^{rel}(\alpha X, \alpha Y, \beta X', \gamma Z' R) = A_\lambda^{rel}(X, Y, X', Z') \quad \forall \alpha, \beta, \gamma > 0 \quad \text{and} \quad RR^\top = R^\top R = I$$

Note that for each $A \in \mathbb{R}^{d_X \times d_Y}$, we have

$$\frac{\mathbb{E}[\|\alpha X A - \alpha Y\|^2]}{\operatorname{Tr}(\operatorname{cov}(\alpha Y))} = \frac{\alpha^2 \, \mathbb{E}[\|XA - Y\|^2]}{\alpha^2 \operatorname{Tr}(\operatorname{cov}(Y))} = \frac{\mathbb{E}[\|XA - Y\|^2]}{\operatorname{Tr}(\operatorname{cov}(Y))}$$

And

$$\begin{aligned}
\frac{\|\operatorname{cov}(\beta X' A, \gamma Z' R)\|_F^2}{\left(\frac{d_Y}{d_X}\right) \operatorname{Tr}(\operatorname{cov}(\beta X')) \operatorname{Tr}(\operatorname{cov}(\gamma Z' R))} &= \frac{\beta^2 \gamma^2 \|\operatorname{cov}(X' A, Z' R)\|_F^2}{\left(\frac{d_Y}{d_X}\right) \beta^2 \gamma^2 \operatorname{Tr}(\operatorname{cov}(X')) \operatorname{Tr}(\operatorname{cov}(Z' R))} \\
&= \frac{\|R^\top \operatorname{cov}(X' A, Z')\|_F^2}{\left(\frac{d_Y}{d_X}\right) \operatorname{Tr}(\operatorname{cov}(X')) \operatorname{Tr}(R^\top \operatorname{cov}(Z') R)} \\
&= \frac{\|\operatorname{cov}(X' A, Z')\|_F^2}{\left(\frac{d_Y}{d_X}\right) \operatorname{Tr}(\operatorname{cov}(X')) \operatorname{Tr}(R R^\top \operatorname{cov}(Z'))} \\
&= \frac{\|\operatorname{cov}(X' A, Z')\|_F^2}{\left(\frac{d_Y}{d_X}\right) \operatorname{Tr}(\operatorname{cov}(X')) \operatorname{Tr}(\operatorname{cov}(Z'))}
\end{aligned}$$

Hence for each $\lambda \geq 0$, the same objective is being minimized is both cases, which proves the invariance.

4.6.3.2 Invariance with respect to data repetition

Let $A_\lambda^{rel}(X, Y, X', Z') = A_\lambda^{rel}$. We need to show that for each $\lambda \geq 0$ and $\forall n, m, l \in \mathbb{N}$:

$$A_\lambda^{rel}\left(\underbrace{(X,\cdots,X)}_{n},\underbrace{(Y,\cdots,Y)}_{m},\underbrace{(X',\cdots,X')}_{n},\underbrace{(Z',\cdots,Z')}_{l}\right)=\left[\begin{array}{c|c|c|c} A_\lambda^{rel} & A_\lambda^{rel} & \cdots & A_\lambda^{rel} \\ \hline 0 & 0 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & 0 \end{array}\right] \tag{4.36}$$

Here, the matrix on the right, which we denote by $\mathcal{A}^*$, is of size $nd_X \times md_Y$ (each block is of size $d_X \times d_Y$).

Note that

$$\frac{\mathbb{E}[\|\overbrace{(X,\cdots,X)}^{n}\mathcal{A}^*-\overbrace{(Y,\cdots,Y)}^{m}\|^2]}{\mathrm{Tr}(\mathrm{cov}(\underbrace{(Y,\cdots,Y)}_{m}))}=\frac{m\,\mathbb{E}[\|XA_\lambda^{rel}-Y\|^2]}{m\,\mathrm{Tr}(\mathrm{cov}(Y))}=\frac{\mathbb{E}[\|XA_\lambda^{rel}-Y\|^2]}{\mathrm{Tr}(\mathrm{cov}(Y))} \tag{4.37}$$

Also,

$$\begin{aligned}\frac{\|\mathrm{cov}(\overbrace{(X',\cdots,X')}^{n}\mathcal{A}^*,\overbrace{(Z',\cdots,Z')}^{l})\|_F^2}{\left(\frac{d_Ym}{d_Xn}\right)\mathrm{Tr}(\mathrm{cov}(\underbrace{(X',\cdots,X')}_{n}))\,\mathrm{Tr}(\mathrm{cov}(\underbrace{(Z',\cdots,Z')}_{l}))}&=\frac{ml\|\mathrm{cov}\left(X'A_\lambda^{rel},Z'\right)\|_F^2}{\left(\frac{d_Ym}{d_Xn}\right)n\,\mathrm{Tr}(\mathrm{cov}(X'))l\,\mathrm{Tr}(\mathrm{cov}(Z'))}\\&=\frac{\|\mathrm{cov}\left(X'A_\lambda^{rel},Z'\right)\|_F^2}{\left(\frac{d_Y}{d_X}\right)\mathrm{Tr}(\mathrm{cov}(X'))\,\mathrm{Tr}(\mathrm{cov}(Z'))}\end{aligned} \tag{4.38}$$

Now note that for any $\mathcal{A}\in\mathbb{R}^{nd_X\times md_Y}$, the estimate $\underbrace{(\widehat{Y,\cdots,Y})}_{m}=\underbrace{(X,\cdots,X)}_{n}\mathcal{A}$ can be written as:

$$\underbrace{(X,\cdots,X)}_{n}\left[\begin{array}{c|c|c|c} A_1 & A_2 & \cdots & A_m \\ \hline 0 & 0 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & 0 \end{array}\right] \tag{4.39}$$

for some matrices $A_1,\cdots,A_m$ of size $d_X\times d_Y$. Therefore, the optimization problem over $\mathcal{A}\in\mathbb{R}^{nd\times mp}$ to find $A_\lambda^{rel}\left(\underbrace{(X,\cdots,X)}_{n},\underbrace{(Y,\cdots,Y)}_{m},\underbrace{(X',\cdots,X')}_{n},\underbrace{(Z',\cdots,Z')}_{l}\right)$

boils down to the optimization problem over block matrices of type in (4.39). Therefore, for all $\lambda \geq 0$, the objective below:

$$\frac{\mathbb{E}[\| \overbrace{(X, \cdots, X)}^{n} \mathcal{A} - \overbrace{(Y, \cdots, Y)}^{m} \|^2]}{\operatorname{Tr}(\operatorname{cov}(\underbrace{(Y, \cdots, Y)}_{m}))} + \lambda \frac{d_X n \|\operatorname{cov}(\overbrace{(X', \cdots, X')}^{n} \mathcal{A}, \overbrace{(Z', \cdots, Z')}^{l})\|_F^2}{d_Y m \operatorname{Tr}(\operatorname{cov}(\underbrace{(X', \cdots, X')}_{n})) \operatorname{Tr}(\operatorname{cov}(\underbrace{(Z', \cdots, Z')}_{l}))}$$

is equal to

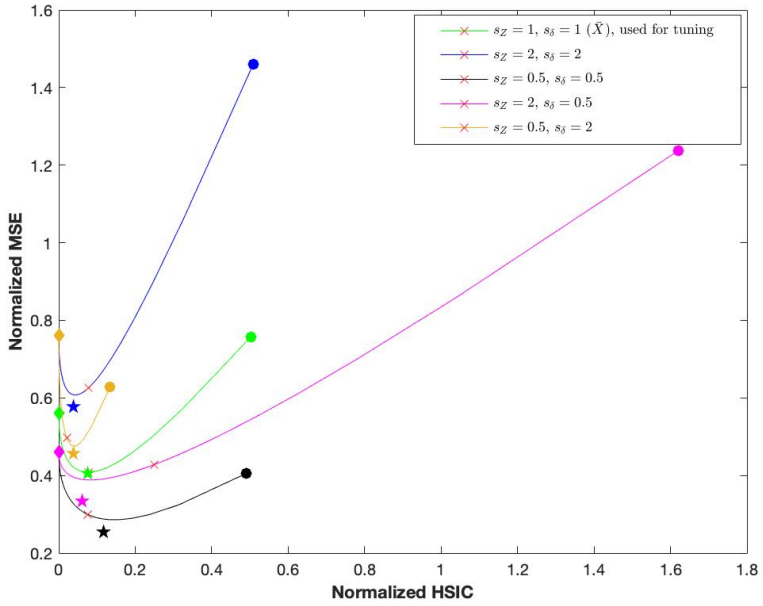
$$\frac{\sum_{i=1}^{m} \mathbb{E}[\|XA_i - Y\|^2]}{m \operatorname{Tr}(\operatorname{cov}(Y))} + \lambda \frac{\sum_{i=1}^{m} l \|\operatorname{cov}(X'A_i, Z')\|_F^2}{\left(\frac{d_Y m}{d_X n}\right) n \operatorname{Tr}(\operatorname{cov}(X')) l \operatorname{Tr}(\operatorname{cov}(Z'))}$$

and has the lower bound of $\frac{\mathbb{E}[\|XA_\lambda^{rel} - Y\|^2]}{\operatorname{Tr}(\operatorname{cov}(Y))} + \lambda \frac{\|\operatorname{cov}\left(X'A_\lambda^{rel}, Z'\right)\|_F^2}{\left(\frac{d_Y}{d_X}\right) \operatorname{Tr}(\operatorname{cov}(X')) \operatorname{Tr}(\operatorname{cov}(Z'))}$ because of the definition of A_λ^{rel}, and since it achieves this lower bound at $\mathcal{A} = \mathcal{A}^*$ as we showed in (4.37) and (4.38), proof of (4.36) is complete.

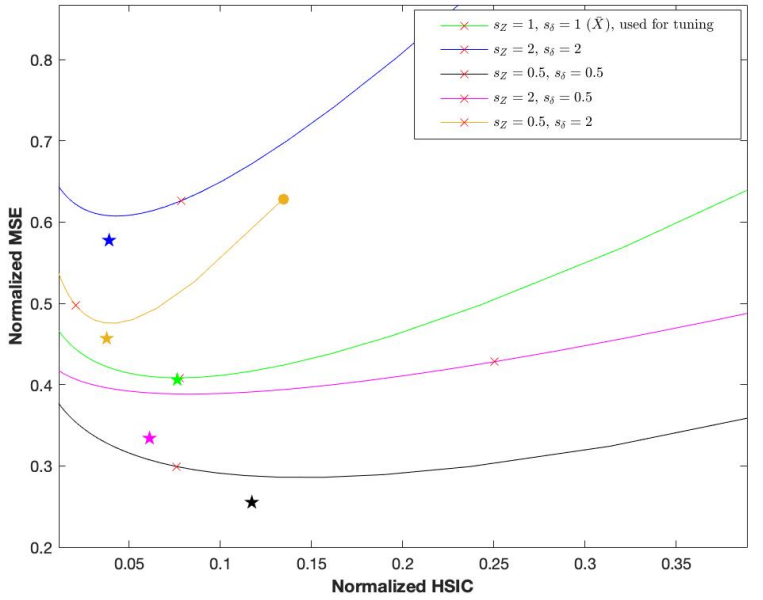
4.6.4 Effectiveness of hyper-parameter tuning (Remark 5)

We consider example (B) from the results section, which uses a high-dimensional realistic calibration setting.

Figure 4.2 presents the relative Mean Squared Error (MSE) with respect to the relative covariance term for different values of scaling factors $s_Z, s_\delta \in \{0.5, 2\}$. Each plot shows the performance of various decoders (cf. section (I)): 1) Optimal linear estimator (OLE) decoder (A_0, b_0): represented by a dot marker, 2) Hypothetical optimal decoder $(A_{opt}^{test}, b_{opt}^{test})$: represented by a star marker, 3) Uncorrelated decoder (A_∞, b_∞) represented by a diamond marker, and 5) Regularized decoders (A_λ, b_λ) for varying values of the hyper-parameter λ, represented by lines. The red "x" marker on each plot highlights the decoder associated with the tuned hyper-parameter λ^* $(A_{\lambda^*}, b_{\lambda^*})$. Figure 4.2b provides a zoomed-in view of Figure 4.2.



(a) Relative MSE of decoders versus relative HSIC, for varying scaling factors s_Z and s_δ



(b) Figure 4.2a zoomed in.

Figure 4.2: Hyper-parameter tuning effectiveness

As shown in the figures, for most settings where (s_Z, s_δ) deviates from $\bar{X}$, the performance of the decoder using the tuned hyper-parameter λ^* closely approaches the minimum achievable performance using regularization, which is denoted by $(A_{\lambda_{min}}, b_{\lambda_{min}})$ in the results section. This observation validates the effectiveness of our heuristic approach

for hyper-parameter tuning.

Remark 8. As evident from Figure 4.2, the uncorrelated decoder exhibits strong robustness to gesture variations. This aligns with its design principle: to minimize the influence of secondary tasks (gestures) on the decoded primary task (movement). In other words, for a constant noise scaling factor (s_δ), changing the gesture scaling factor (s_Z) does not affect the uncorrelated decoder's mean squared error (MSE). This robustness can be analytically shown:

Proof. In this simulation, both Σ_X and $\Sigma_{Z'X'}$ are of full rank, therefore, from corollary 14, $A_\infty = H(H^\top \Sigma_X H)^{-1} H^\top \Sigma_{XY}$ for $H \in \mathbb{R}^{d_X \times (d_X - d_Z)}$ with column space equal to null$(\Sigma_{Z'X'})$. Therefore, by equation (4.16), the MSE for A_∞ can be written as below:

$$\begin{aligned}\mathbb{E}[\|X^{test} A_\infty - Y\|^2] &= \mathrm{Tr}\left(A_\infty^\top \Sigma_{X^{test}} A_\infty - 2A_\infty^\top \Sigma_{X^{test}Y} + \Sigma_Y\right) \\ &= \mathrm{Tr}\left(\Sigma_{YX} H(H^\top \Sigma_X H)^{-1} H^\top \Sigma_{X^{test}} H(H^\top \Sigma_X H)^{-1} H^\top \Sigma_{XY}\right) \\ &\quad - 2\left(\Sigma_{YX} H(H^\top \Sigma_X H)^{-1} H^\top \Sigma_{X^{test}Y}\right) + \mathrm{Tr}(\Sigma_Y)\end{aligned} \tag{4.40}$$

Now note that from the model $X^{test} = YU + s_Z Z'V + s_\delta \delta$, it follows that:

$$\Sigma_{X^{test}} = U^\top \Sigma_Y U + s_Z^2 V^\top \Sigma_{Z'} V + s_\delta^2 \Sigma_\delta = U^\top \Sigma_Y U + s_Z^2 V^\top \Sigma_{Z'X'} + s_\delta^2 \Sigma_\delta$$

, and

$$\Sigma_{X^{test}Y} = U^\top \Sigma_Y = \Sigma_{XY}$$

Therefore, for a fixed s_δ and a varying s_Z, the last two traces in (4.40) are fixed, and because of the property of H, the first trace can be written as below:

$$\mathrm{Tr}\left(\Sigma_{YX} H(H^\top \Sigma_X H)^{-1} H^\top (U^\top \Sigma_Y U + s_\delta^2 \Sigma_\delta) H(H^\top \Sigma_X H)^{-1} H^\top \Sigma_{XY}\right)$$

which is also fixed. □

From this proof, we can see that not only does scaling the gesture influence leave the MSE unchanged, but any change to the influence of the secondary task that falls within the column space of $\Sigma_{Z'X'}$ will also produce the same MSE for the uncorrelated decoder. It is important to note that in the context of Figure 4.2, both training models remained unchanged. We only varied the scaling factors for the test model, which is why the uncorrelated decoder matrix (A_∞) is the same across all examples in the figure. However, any equivalent change in the user's gestures during gesture calibration would also lead to the same uncorrelated decoder matrix (A_∞), assuming all other characteristics of both training models remain constant. This further reinforces this decoder's robustness to gesture variations during both training and test phase.

4.6.4.1 Derivation of (4.10)

For the feature-centered case where $\mathbb{E}[X] = \mathbb{E}[Y] = 0$ (for simplicity), the tuned hyper-parameter λ^* in equation (4.9) in the main text boils down to

$$\lambda^* = \arg\min_{\lambda \geq 0} \mathbb{E}[\|Y - (\bar{X}A_\lambda)\|^2] \tag{4.41}$$

Let $f(\lambda) = \mathbb{E}[\|Y - (\bar{X}A_\lambda)\|^2]$. Using equations (4.16) and (4.20) we have

$$\begin{aligned}
\lambda^* &= \arg\min_{\lambda \geq 0} f(\lambda) \\
&= \arg\min_{\lambda \geq 0} \mathbb{E}[\|Y - (\bar{X}A_\lambda)\|^2] \\
&= \arg\min_{\lambda \geq 0} \operatorname{Tr}\left(A_\lambda^\top \Sigma_{\bar{X}} A_\lambda - 2A_\lambda^\top \Sigma_{\bar{X}Y} + \Sigma_Y\right) \\
&= \arg\min_{\lambda \geq 0} \operatorname{Tr}\left(\Sigma_{YX} M_\lambda \Sigma_{\bar{X}} M_\lambda \Sigma_{XY} - 2\Sigma_{YX} M_\lambda \Sigma_{\bar{X}Y}\right) \\
&= \arg\min_{\lambda \geq 0} g(M_\lambda)
\end{aligned} \tag{4.42}$$

where $M_\lambda = (\Sigma_X + \lambda \Sigma_{X'Z'} \Sigma_{Z'X'})^{-1}$ (which is symmetric), and

$$g(M_\lambda) = \operatorname{Tr}\left(\Sigma_{YX} M_\lambda \Sigma_{\bar{X}} M_\lambda \Sigma_{XY} - 2\Sigma_{YX} M_\lambda \Sigma_{\bar{X}Y}\right)$$

.

4.6.5 Derivations of remark 4

We consider two cases for Y:

1. Y **is continuous**:

   For this case, the loss function we consider is the squared error loss: $L(\widehat{Y}, Y) = \|\widehat{Y} - Y\|^2$ where $\widehat{Y} = f(XA + b, A, b)$. One can define f as below

$$f(XA + b, A, b) \triangleq \mathbb{E}[Y|XA + b] \tag{4.43}$$

   Note that given the squared error loss, this definition makes sense as we have

$$\text{argmin}_h \mathbb{E}[L(h(XA + b), Y)] = \mathbb{E}[Y|XA + b] \tag{4.44}$$

   Now the definition of $(A^f_\lambda, b^f_\lambda)$ in (4.6) in the main text becomes

$$(A^f_\lambda, b^f_\lambda) \triangleq \underset{(A,b)}{\arg\min}\, \mathbb{E}[\|Y - \mathbb{E}[Y|XA + b]\|^2] + \lambda \|\text{cov}(X'A, Z')\|^2_F \tag{4.45}$$

   - **Analytical equivalent objective function**: The objective function in (4.45) can be simplified to have a closed-form equivalent, which can be fed into an optimizer to find $(A^f_\lambda, b^f_\lambda)$. To achieve this, we first need to find $\mathbb{E}[Y|XA+b]$. Note that under the linear Gaussian assumption we have already made for X and Y (described in Remark 3 in the main text), Y and $W = XA + b$ are also jointly multivariate Gaussian, and their joint partitioned covariance matrix Σ and mean μ maintains the form below:

$$\Sigma = \left[\begin{array}{c|c} \Sigma_{Y,Y} & \Sigma_{Y,W} \\ \hline \Sigma_{W,Y} & \Sigma_{W,W} \end{array}\right] = \left[\begin{array}{c|c} \Sigma_Y & \Sigma_{YX}A \\ \hline A^\top \Sigma_{XY} & A^\top \Sigma_X A \end{array}\right]_{2p \times 2p} \tag{4.46}$$

and

$$\mu = \left[\begin{array}{c|c} \mu_Y & \mu_X A + b \end{array} \right]_{1\times 2p} \tag{4.47}$$

Then we have (see [117] for details)

$$\begin{aligned} f(XA+b, A, b) &= \mathbb{E}[Y|XA+b] \\ &= \mu_Y + (XA+b-(\mu_X A+b))\Sigma_{W,W}^{-1}\Sigma_{W,Y} \\ &= \mu_Y + (X-\mu_X)A(A^\top \Sigma_X A)^{-1}A^\top \Sigma_{XY} \end{aligned} \tag{4.48}$$

The definition (4.45) then becomes

$$\begin{aligned} (A_\lambda^f, b_\lambda^f) &= \arg\min_{(A,b)} \mathbb{E}[\|Y - \mathbb{E}[Y|XA+b]\|^2] + \lambda \|\text{cov}(X'A, Z')\|_F^2 \\ &= \arg\min_{(A,b)} \mathbb{E}[\|Y - \mu_Y - (X-\mu_X)A(A^\top\Sigma_X A)^{-1}A^\top\Sigma_{XY}\|^2] \\ &\qquad + \lambda\|\text{cov}(X'A, Z')\|_F^2 \\ &= (\arg\min_{A} \mathbb{E}[\|(Y-\mu_Y) - (X-\mu_X)A(A^\top\Sigma_X A)^{-1}A^\top\Sigma_{XY}\|^2] \\ &\qquad + \lambda\|\text{cov}(X'A, Z')\|_F^2, \tilde{b}) \end{aligned} \tag{4.49}$$

Here, $\tilde{b}$ can be any real vector. For simplicity, we can assume that $b_\lambda^f = 0$. (4.49) shows that $(A_\lambda^f, b_\lambda^f)$ is the same for zero-mean random vectors X and Y. So we can simplify the expectation term for feature-centered case $\mathbb{E}[L(f(XA+b, A, b), Y)] = \mathbb{E}[\|Y - XA(A^\top\Sigma_X A)^{-1}A^\top\Sigma_{XY}\|^2]$ as below

$$
\begin{aligned}
&\mathbb{E}[\mathrm{Tr}((Y - XA(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY})^\top (Y - XA(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY}))] \\
&= \mathbb{E}[\mathrm{Tr}(\Sigma_{YX} A(A^\top \Sigma_X A)^{-1} A^\top X^\top XA(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY}) \\
&- 2\,\mathrm{Tr}(XA(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY} Y) + \mathrm{Tr}(Y^\top Y)] \\
&= \mathrm{Tr}(\mathbb{E}[\Sigma_{YX} A(A^\top \Sigma_X A)^{-1} A^\top X^\top XA(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY}]) \\
&- 2\,\mathrm{Tr}(\mathbb{E}[XA(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY} Y]) + \mathrm{Tr}(\mathbb{E}[Y^\top Y]) \\
&= \mathrm{Tr}(\Sigma_{YX} A(A^\top \Sigma_X A)^{-1} A^\top \Sigma_X A(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY}) \\
&- 2\,\mathrm{Tr}(A(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY} \Sigma_{YX}) + \mathrm{Tr}(\Sigma_Y) \\
&= \mathrm{Tr}(\Sigma_{YX} A(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY}) \\
&- 2\,\mathrm{Tr}(\Sigma_{YX} A(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY}) + \mathrm{Tr}(\Sigma_Y) \\
&= \mathrm{Tr}(\Sigma_Y - \Sigma_{YX} A(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY})
\end{aligned}
$$

Therefore, by adding the penalty term in (4.17), one can fed

$$
\mathrm{Tr}(\Sigma_Y - \Sigma_{YX} A(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY} A^\top \Sigma_{Z'X'}^\top \Sigma_{Z'X'} A) \tag{4.50}
$$

to an optimizer to find A_λ^f, or solve the following equation (4.51) for $\frac{\partial}{\partial A}\,\mathbb{E}[L(f(XA+b, A, b), Y)]$ as below

$$
-2\Sigma_X A(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY} \Sigma_{YX} A(A^\top \Sigma_X A)^{-1} + 2\Sigma_{XY} \Sigma_{YX} A(A^\top \Sigma_X A)^{-1}
$$

(see [113]). Therefore, adding the derivative of the covariance term, A_λ^f is the solution of the equation below

$$
\begin{aligned}
&- 2\Sigma_X A(A^\top \Sigma_X A)^{-1} A^\top \Sigma_{XY} \Sigma_{YX} A(A^\top \Sigma_X A)^{-1} \\
&+ 2\Sigma_{XY} \Sigma_{YX} A(A^\top \Sigma_X A)^{-1} \\
&+ 2\lambda \Sigma_{Z'X'}^\top \Sigma_{Z'X'} A \\
&= 0
\end{aligned} \tag{4.51}
$$

- **Applying f on (A, b)'s derived from previous approaches**: Figure 4.3

shows the relative MSE of decoder $\widehat{Y} = f(XA_\lambda + b_\lambda, A_\lambda, b_\lambda)$ for $\lambda \in [0, 10] \cup \{\infty\}$, and $\widehat{Y} = f(XA_{opt}^{test} + b_{opt}^{test}, A_{opt}^{test}, b_{opt}^{test})$ for the setting of example 1 (scenario of concurrent move and gesture at test time) from the results section. For each specific (A, b), we also show the relative MSE of the corresponding $\widehat{Y} = XA + b$. Note that for $f(XA + b, A, b) = \mathbb{E}[Y|XA + b]$ in this figure, we are not using $(A_\lambda^f, b_\lambda^f)$ in (4.49), in other words, we neither optimize (4.50) nor find the solution of (4.51), rather, we employ already computed (A, b)'s from different approaches (see section (I) of the main text for their description), and for each (A, b), we find its corresponding $f(XA + b, A, b) = \mathbb{E}[Y|XA + b]$ from (4.48).

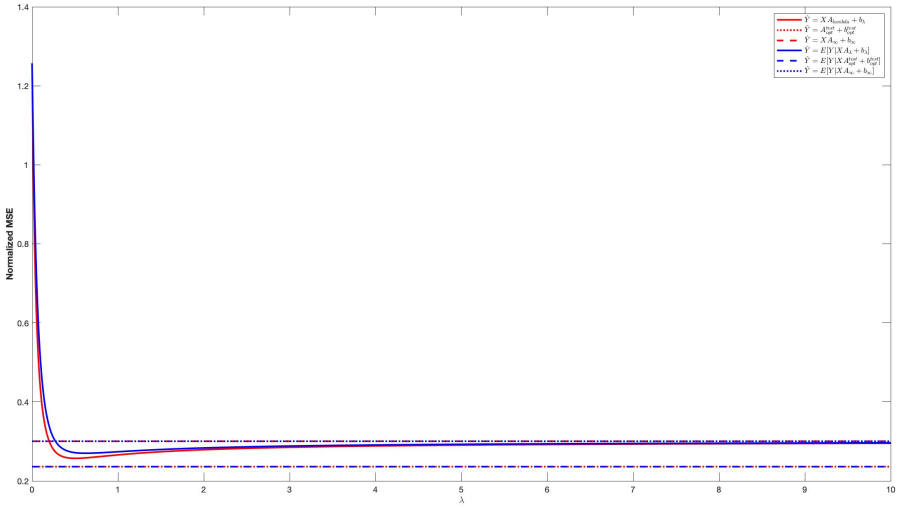


Figure 4.3: Relative MSE versus λ, for decoders $\widehat{Y} = XA + b$ and $\widehat{Y} = \mathbb{E}[Y|XA + b]$ for different (A, b)'s, for test scenario of concurrent Y and Z of example 1.

Remark 9. Note that Figure 4.3 demonstrates a realistic scenario where the training distribution (which is used to train any decoder, including $\mathbb{E}[Y|XA + b]$ from equation (4.48)) is different from the test distribution, and the MSE's are shown for test data. This explains why $\widehat{Y} = f(XA + b, A, b)$ is not performing better than $\widehat{Y} = XA + b$ for a fixed (A, b). In fact, if there was no distribution shift between train and test, then by equation (4.44), $f(XA + b, A, b)$ would have had less MSE than $XA + b$ for any A.

Remark 10. As can be observed from the Figure 4.3, we can show that for both $A \in \{A_\infty, A_0, A_{opt}^{test}\}$, we have $XA = \mathbb{E}[Y|XA]$:

$$(A_0^\top \Sigma_X A_0)^{-1} A_0^\top \Sigma_{XY} \overset{(4.20)}{=} (\Sigma_{YX}\Sigma_X^{-1}\Sigma_X\Sigma_X^{-1}\Sigma_{XY})^{-1}\Sigma_{YX}\Sigma_X^{-1}\Sigma_{XY} = I$$

Note that A_{opt}^{test} is nothing but the same equality above, except every term is from the test distribution. Also, by (4.32), $(A_\infty^\top \Sigma_X A_\infty)^{-1} A_\infty^\top \Sigma_{XY}$ is equal to

$$\begin{aligned}
&(\Sigma_{YX}H(H^\top\Sigma_X H)^{-1}H^\top\Sigma_X H(H^\top\Sigma_X H)^{-1}H^\top\Sigma_{XY})^{-1}A_\infty^\top\Sigma_{XY} \\
&= (\Sigma_{YX}H(H^\top\Sigma_X H)^{-1}H^\top\Sigma_{XY})^{-1}\Sigma_{YX}H(H^\top\Sigma_X H)^{-1}H^\top\Sigma_{XY} \\
&= I
\end{aligned}$$

Now since by (4.48), $\mathbb{E}[Y|XA] = XA(A^\top\Sigma_X A)^{-1}A^\top\Sigma_{XY}$, we are done.

2. Y **is discrete**:

For this case, the loss function we consider is $L(\widehat{Y}, Y) = \mathbb{1}_{\widehat{Y}\neq Y}$, and we define f as below:

$$f(XA + b, A, b) \triangleq \text{argmax}_y \, \mathbb{P}(Y = y|XA + b) \tag{4.52}$$

Given the squared error loss, this definition makes sense since $\mathbb{E}[L(h(XA + b), Y)] = \mathbb{P}(h(XA + b) \neq Y)$ and, over h, it is minimized when $h(XA + b) = f(XA + b, A, b)$ as above.

Therefore, the definition of $(A_\lambda^f, b_\lambda^f)$ in (4.6) in the main text is as follows:

$$(A_\lambda^f, b_\lambda^f) \triangleq \underset{(A,b)}{\arg\min}\, \mathbb{P}\left(Y \neq f(XA + b, A, b)\right) + \lambda\|\text{cov}(X'A, Z')\|_F^2 \tag{4.53}$$

- **Analytical equivalent objective function**: For simplicity, we assume that Y takes two values, i.e., $Y \in \{m, n\} \subseteq \mathbb{R}^{1\times d_Y}$, and $\mathbb{P}(Y = n) = p$. Note that from the linear Gaussian encoding model assumption for the primary calibration dataset $X = YU + \epsilon$ (see Remark 3 in the main text), we have:

$$
\begin{aligned}
XA + b \mid Y = m &\sim N_{d_Y}(mUA + b, A^\top \Sigma_\epsilon A) \\
XA + b \mid Y = n &\sim N_{d_Y}(nUA + b, A^\top \Sigma_\epsilon A)
\end{aligned}
\tag{4.54}
$$

Then the first term in (4.53), i.e., $\mathbb{P}\left(Y \neq f(XA+b, A, b)\right)$ can be simplified as below:

$$
\begin{aligned}
&\mathbb{P}\left(f(XA + b, A, m) = n \mid Y = m\right)\mathbb{P}(Y = m) \\
&+ \mathbb{P}\left(f(XA + b, A, b) = m \mid Y = n\right)\mathbb{P}(Y = n) \\
&= (1 - p)\,\mathbb{P}\Big(\mathbb{P}\left(Y = n \mid XA + b\right) \geq \mathbb{P}\left(Y = m \mid XA + b\right) \mid Y = m\Big) \\
&+ p\,\mathbb{P}\Big(\mathbb{P}\left(Y = m \mid XA + b\right) \geq \mathbb{P}\left(Y = n \mid XA + b\right) \mid Y = n\Big)
\end{aligned}
\tag{4.55}
$$

Let $M = \mathbb{P}\left(\mathbb{P}\left(Y = n \mid XA + b\right) \geq \mathbb{P}\left(Y = m \mid XA + b\right) \mid Y = m\right)$ and $N = \mathbb{P}\left(\mathbb{P}\left(Y = m \mid XA + b\right) \geq \mathbb{P}\left(Y = n \mid XA + b\right) \mid Y = n\right)$. Using (4.54) we have

$$
\begin{aligned}
M = \mathbb{P}_{W_m}\Big(&p\frac{1}{\sqrt{\det(A^\top \Sigma_\epsilon A)}} \\
&\exp\big(-\frac{1}{2}(w - nUA - b)(A^\top \Sigma_\epsilon A)^{-1}(w - nUA - b)^\top\big) \geq \\
&(1 - p)\frac{1}{\sqrt{\det(A^\top \Sigma_\epsilon A)}} \\
&\exp\big(-\frac{1}{2}(w - mUA - b)(A^\top \Sigma_\epsilon A)^{-1}(w - mUA - b)^\top\big)\Big)
\end{aligned}
\tag{4.56}
$$

where $W_m \sim N_{d_Y}(mUA + b, A^\top \Sigma_\epsilon A)$.

Let $S = (A^\top \Sigma_\epsilon A)^{-1}$, $\mu_m = mUA + b$, $\mu_n = nUA + b$, then

$$
\begin{aligned}
M &= \mathbb{P}_{W_m}\Big(-\frac{1}{2}\big((w - \mu_m)S(w - \mu_m)^\top - (w - \mu_n)S(w - \mu_n)^\top\big) \geq \\
&\quad \ln(\frac{1 - p}{p})\Big) \\
&= \mathbb{P}_{W_m}\Big(2w(S\mu_n^\top - S\mu_m^\top) + \mu_m S\mu_m^\top - \mu_n S\mu_n^\top \leq 2\ln(\frac{1 - p}{p})\Big) \\
&= \Psi_m(2\ln(\frac{1 - p}{p}) - \mu_m S\mu_m^\top + \mu_n S\mu_n^\top)
\end{aligned}
\tag{4.57}
$$

where $\Psi_m(t)$ represents the cdf of one-dimensional Gaussian distribution of $2W_mS(\mu_n^\top - \mu_m^\top)$ at point t. Note that $2W_mS(\mu_n^\top - \mu_m^\top)$ is distributed as

$$N_1\big(2(mUA+b)S(\mu_n^\top - \mu_m^\top), 4(\mu_n - \mu_m)S^\top A^\top \Sigma_\epsilon A S(\mu_n^\top - \mu_m^\top)\big)$$

or,

$$N_1\big(2(mUA+b)(A^\top\Sigma_\epsilon A)^{-1}A^\top U^\top(n^\top - m^\top) \\ , 4(n-m)UA(A^\top\Sigma_\epsilon A)^{-1}A^\top U^\top(n^\top - m^\top)\big)$$

Similarly, $N = 1 - \Psi_n(2\ln(\frac{1-p}{p}) - \mu_m S\mu_m^\top + \mu_n S\mu_n^\top)$ can be found, where Ψ_n is the cdf of $2W_nS(\mu_n^\top - \mu_m^\top)$ where $W_n \sim N_{d_Y}(nUA+b, A^\top\Sigma_\epsilon A)$. Therefore, (4.55) is found, and adding the covariance term in (4.17), can be fed to an optimizer to find $(A_\lambda^f, b_\lambda^f)$.

- **Applying f on (A,b)'s derived from previous approaches**: Figure 4.4 shows the 0-1 loss for the decoder $f(XA_\lambda + b_\lambda, A_\lambda, b_\lambda)$ defined in (4.52) for varying $\lambda \in [0,50] \cup \{\infty\}$, where (A_λ, b_λ)'s represent the linear decoder derived from (4.20), besides the 0-1 loss of the decoder $f(XA_{opt}^{test} + b_{opt}^{test}, A_{opt}^{test}, b_{opt}^{test})$. We also defined another decoder as below:

$$g(XA+b, A, b) \triangleq \text{argmin}_y \|y - (XA+b)\| \tag{4.58}$$

And we include the performance of this decoder for the same (A,b)'s. The example used for this figure is a data-rich environment where $d_X = 300$, $d_Y = 2$, $d_Z = 4$, and Y takes two randomly chosen vectors each with probability 0.5.

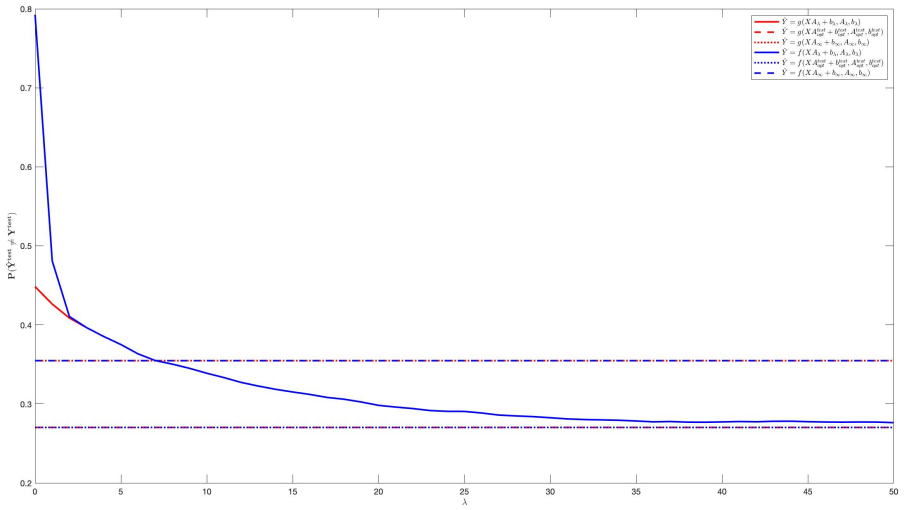


Figure 4.4: Performance of two different non-linear decoders $\widehat{Y} = f(XA + b, A, b)$ and $\widehat{Y} = g(XA + b, A, b)$ for continuous Y, applied on different (A, b)'s.

CHAPTER 5

Developing globally representative GEDI waveform-to-biomass models using crossover data

5.1 Introduction

NASA's Earth System Science Pathfinder (ESSP) program includes the Global Ecosystem Dynamics Investigation (GEDI), that was launched to the International Space Station (ISS) in December 2018. One of GEDI's three key goals is to quantify the spatial distribution of aboveground carbon in Earth's tropical and temperate forests. Accurate estimation of GEDI footprint aboveground biomass density (AGBD) is crucial for implementing forest conservation policies that mitigate climate change, such as the United Nation's Reduced Emissions from Deforestation and forest Degradation (REDD+) program [44, 118–120], and reducing uncertainties in global carbon stocks [45, 121, 122].

GEDI, a multi-beam laser altimeter, uses returned lidar waveforms to quantify the vertical distribution of vegetation after geolocating them. This processing involves identifying the ground surface [123]. Subsequently, the footprint-level canopy height, canopy

vertical profile and Relative Height (RH) energy metrics are calculated [124, 125]. These metrics are stored in the GEDI02_A product and are then used, leveraging statistical models, to estimate AGBD at the footprint level, resulting in the GEDI04_A product [45]. An ideal approach for estimating AGBD from GEDI data involves calibrating models using a comprehensive dataset that includes both on-orbit GEDI measurements and corresponding field estimates of AGBD. However, such datasets are often limited in availability, making this approach challenging [45, 46]. To address this limitation, researchers have leveraged simulated GEDI waveforms derived from discrete-return airborne lidar (ALS) data [47, 48]. These simulations, typically conducted over ecologically significant forests with existing field measurements, provide a substitute for real GEDI data and enable the development of globally representative models for AGBD prediction. The most geographically comprehensive dataset of this type is the Forest Structure and Biomass Database (FSBD) [46]. This quality-filtered dataset contains 8,549 pairs of height metrics derived from simulated GEDI waveforms, along with corresponding field estimates of AGBD measured at a footprint size comparable to GEDI [126].

Numerous studies have employed height metrics derived from discrete-return airborne lidar (ALS) to predict AGBD using various statistical models across diverse spatial resolutions, scales, and ecosystems [127–143] (See [144] for a comprehensive review). However, this approach introduces a key challenge: ensuring the transferability of models developed using simulated GEDI waveforms to actual GEDI data.

To address this, we propose a methodology that leverages a large dataset of presumably co-located real and simulated GEDI RH metrics [145]. We will refer to this dataset as the "crossove" dataset. This method assumes certain features for this dataset, including co-location of real and simulated GEDI waveforms, and a temporal overlap between on-orbit GEDI and ALS data collection to ensure unchanged forest structure. From this, we can assume that any additional information provided by real GEDI at a certain footprint beyond that contained in the simulated data is irrelevant for predicting AGBD (Conditional Independence Assumption (CIA), see Methods section).

We exploit this CIA to develop a novel sequential approach for AGBD prediction that utilizes both the training data (FSBD) and the crossover dataset. In essence, we build a model to predict AGBD from simulated GEDI waveforms using the training data. We then apply this model to predict AGBD values for the simulated GEDI waveforms in the crossover dataset, assuming negligible error in the first stage estimation. Finally, we build a second model to map the actual GEDI waveforms in the crossover dataset to the corresponding predicted AGBD values obtained from the first stage. This sequential approach is mathematically justified, strengthening the foundation of our methodology in the Methods section.

To address the potential limitations of the CIA underlying the sequential algorithm, we propose a second approach (single-stage estimation). This approach focuses on improving the transferability of models from simulated to real GEDI data by directly optimizing for consistency in AGBD predictions across both datasets. By incorporating consistency of footprint-level AGBD prediction on this crossover dataset alongside accuracy on the FSBD calibration dataset, we can develop models that are more robust to the transferability challenge. We demonstrate (see the Methods section) that under specific model assumptions, the mean squared error (MSE) between predicted AGBD and actual GEDI RH metrics can be decomposed into two components. The first component is the MSE of the AGBD predicted using the ALS data. The second component, which we term the "consistency term", represents the mean squared difference between the AGBD values predicted from ALS and GEDI data. This theoretical result strengthens the rationale for employing an objective function that combines the prediction error on the FSBD data (using ALS) with the consistency term derived from the crossover dataset.

Previous studies using remote sensing data like lidar for biomass estimation often focused on small geographic domains due to the spatial limitation of the data acquisition [144]. Reported prediction accuracy of those studies that were conducted on larger areas, such as regional, continental, pantropical or global is generally lower than local studies [132, 134, 146, 147]. Likewise, the results presented in this paper are averaged over the globally comprehensive training dataset, acknowledging the potential trade-off between

broader coverage and reduced accuracy.

5.2 Mathematical framework

This section formulates the main question this paper addresses and summarizes the proposed solutions and underlying assumptions. Detailed methodologies are provided in the Methods section.

For a specific footprint, let Y, X, and X' denote the feature to be predicted (AGBD in this case), the covariate features (e.g. RH metrics) of the real GEDI waveform used to predict Y, and the corresponding covariate features of the simulated GEDI, respectively. Here, X and X' belong to the same feature space (e.g., $\mathbb{R}^d$, where d is the number of covariate features). The joint distribution between Y, X, and X' is unknown.

The primary objective (producing the GEDI04_A product) is to predict Y from X. However, there is no available dataset comprising samples of (X, Y) that can be used as a training dataset for this estimation. Instead, we have access to two datasets:

1. A dataset of paired observations of (X', Y) (training dataset).
2. A dataset of paired observations of (X, X'), where each pair is believed to correspond to the same footprint (crossover dataset).

The primary question is whether these two datasets of (X', Y) and (X, X') samples can help achieve the main goal: estimating Y from X. Specifically, the goal is to minimize the mean squared loss $\mathbb{E}[h(X) - Y]$ over h, which is equivalent to learn $r(x) = \mathbb{E}[Y|X = x]$. Generally, the answer is no. However, under certain assumptions outlined below, this goal can be achieved:

(I) **Feature Equivalence** ($X = X'$): Under this idealized scenario, the real and simulated GEDI waveform metrics are identical for the same footprint ($X = X'$). This implies that $p(Y|X) = p(Y|X')$, and the problem reduces to estimating $p(Y|X')$ using only the training dataset. However, this assumption is unrealistic due to factors such as sensor noise in real GEDI waveforms and uncertainties in

the extraction of RH metrics from real GEDI data compared to simulated data [46, 48, 148].

(II) **Posterior Mean Equivalence** ($E[Y|X = x] = E[Y|X' = x]$ for (almost) all x) : One can relax the naive assumption of $X = X'$ by adopting a less restrictive one: for (almost) all $x \in \mathbb{R}^d$, $\mathbb{E}[Y|X = x] = \mathbb{E}[Y|X' = x]$. This implies that while the full conditional distributions might differ, their expected values coincide. In this case, the problem reduces to estimating $E[Y|X' = x]$ on training data.

Special Cases:

- **II-A** Consider the following model:

$$\begin{aligned} Y &= X\gamma + \epsilon \\ X &= X' + e \end{aligned} \tag{5.1}$$

where γ is a coefficient vector, ϵ is a noise term with zero mean independent of X, and e is a noise term with zero mean such that (e, ϵ, X') are mutually independent. Under these assumptions, it can be shown that $\mathbb{E}[Y|X = x] = x\gamma = \mathbb{E}[Y|X' = x]$ for all x.

- **II-B** For the following model

$$\begin{aligned} Y &= g(ZA, \delta) \\ X &= ZA + \epsilon A^{\perp} \\ X' &= ZA + \epsilon' A^{\perp} \end{aligned} \tag{5.2}$$

where A is a projection operator, $A^{\perp}$ is its orthogonal complement, and Z, ϵ, ϵ' and δ are independent. This model implies that the information relevant for predicting Y lies within a shared subspace defined by A, and the differences between real and simulated GEDI waveforms, captured by ϵ and ϵ', are orthogonal to this subspace. Consequently, $\mathbb{E}[Y|X = x] = \mathbb{E}[Y|X' = x] = \mathbb{E}[g(ZA, \delta)|Z = x]$ for all x.

The assumption of perfect feature equivalence (I) or its weaker case, the posterior mean equivalence in (II)) form the basis for a common approach in the field (see [148]) where models trained on simulated GEDI data X' are directly applied to real GEDI data X. This traditional method relies solely on the training dataset. We do not explore this approach further in this paper. Instead, we explore alternative assumptions and modelling strategies that allows for the integration of the crossover dataset, outlined below:

(III) **Conditional Independence Assumption (CIA)** ($p(Y|X, X') = p(Y|X')$): An alternative way to relax the feature equivalence assumption in (I) is to assume that Y and X are conditionally independent given X' (note that $X = X'$ implies this conditional independence). This assumption becomes plausible if we consider that the real and simulated GEDI samples are indeed associated with the same footprint and are recorded under similar conditions, such as the same time frame or with an unchanged forest structure. Since real GEDI measurements are noisy while simulated GEDI data are not, and the measured features in simulated GEDI are more accurate, it is reasonable to assume that, given the simulated GEDI data at a specific footprint, the addition of real GEDI waveform information does not provide any further information necessary to predict AGBD. Therefore, the Conditional Independence Assumption (CIA) holds under these circumstances. It is important to note that the co-location assumption is essential for the validity of CIA. If geo-location errors occur, the simulated GEDI data from a nearby footprint would be insufficient to accurately predict AGBD at a specific footprint. In such cases, the real GEDI waveform data from that exact footprint is also necessary, highlighting the importance of precise co-location for the CIA to hold.

This is the underlying assumption of our first proposed method. We show that the estimator of interest, $r(X) = \mathbb{E}[Y|X]$, can be estimated by sequentially optimizing these two components: 1) $\mathbb{E}[(Y - g(X'))^2]$: the mean squared error (MSE) of predicting Y using $\widehat{Y} = g(X')$ with the training dataset (X', Y), and 2) $\mathbb{E}[(g(X') - r(X))^2]$: the mean squared difference (MSD) between $g(X')$ and

$r(X)$ in the crossover dataset.

We discuss the detailed methodology for learning this model in Section 5.3.3.1.

- **III-A**: Consider the model below:

$$\begin{aligned} Y &= g(X') + \epsilon' \\ X &= X' + e \end{aligned} \tag{5.3}$$

  where ϵ' and e both have mean zero, ϵ' is independent of X', and e is independent of both X' and ϵ'. This configuration represents a specific instance of the CIA outlined in (III).

  We show that under this model, finding the best linear estimator of Y from X, that is, the linear estimator of $\mathbb{E}[Y|X]$, can also be achieved by minimizing the sum of $\mathbb{E}[(h(X') - Y)^2]$ (MSE of the estimator on the training dataset (X', Y)) and $\mathbb{E}[(h(X) - h(X'))^2]$ (MSD between $h(X')$ and $h(X)$ in the crossover dataset) over linear functions h (see (5.5) for details). Although this model is a special case of the model described in (III), which implies that the same sequential approach can be applied, its specific nature allows for a simpler, single-stage procedure for estimating $\mathbb{E}[Y|X]$. This approach will be discussed in detail in Section 5.3.3.2.

The result on estimating $\mathbb{E}[Y|X]$ under this Conditional Independence Assumption (CIA) aligns with the intuition that a good predictor should not only fit the training data accurately but also demonstrate consistency between simulated and real GEDI waveforms.

5.3 Methods

5.3.1 Data

Unfortunately, there is not currently a readily available dataset that directly pairs real GEDI waveforms with corresponding ground measurements of Aboveground Biomass Density (AGBD) within each footprint area covered by a GEDI measurement.

This section details the two alternative datasets utilized in this study for model development and evaluation.

5.3.1.1 Training data

The training dataset used in this study is the same as that utilized by Kellner et al. for their model development [46]. This dataset comprises 8,549 pairs of simulated GEDI (ALS)/AGBD samples, each supplemented with geographic information, including indicators of world regions and dominant plant functional types (PFTs). Detailed information on the construction of these paired samples is described in detail by Duncanson et al. [149]. In essence, the data comes from locations where both ground-based tree measurements and airborne LiDAR surveys were conducted during the leaf-on season. The LiDAR data, collected within two years of the ground measurements, was used to simulate GEDI waveforms [48]. For each GEDI-sized footprint within these locations, allometric equations were employed to estimate AGBD from the collected tree characteristics [126]. To ensure data quality, the collection of pairs of simulated GEDI waveform, estimated AGBD underwent several data filters to remove potential errors. This resulted in a final dataset of 8,549 reliable pairs representing forests across all continents covered by GEDI. Hereafter, we denote this dataset as $(\mathbf{X'}_t, \mathbf{Y}_t)$, where

- $\mathbf{X'}_t \in \mathbb{R}^{N_t \times d}$ represents a matrix whose i-th row contains features such as relative heights and categorical descriptors like PFT and region (all described below) for the i-th sample.
- $\mathbf{Y}_t \in \mathbb{R}^{N_t \times 1}$ is the corresponding ground-estimated AGBD values.

Here, d denotes the total number of selected features used as covariates in the prediction of AGBD.

5.3.1.2 Crossover data

The crossover dataset consists of a substantial number of sample pairs ($N_c = 676,950$) where a simulated GEDI waveform (ALS) is paired with a corresponding real GEDI waveform believed to correspond to approximately the same geographic footprint. To construct this dataset, real GEDI waveform footprints are first geolocated. Then, using the ALS data for that location, a simulated GEDI waveform is created to match the real For a detailed explanation of how the crossover dataset is constructed, please refer to [145]. Throughout this paper, we denote the crossover data as $(\mathbf{X}_c, \mathbf{X'}_c)$. Here,

- $\mathbf{X}_c \in \mathbb{R}^{N_c \times d}$ represents a matrix where each row (indexed by i) contains the real GEDI waveform measurements for the i-th footprint.
- $\mathbf{X'}_c \in \mathbb{R}^{N_c \times d}$ represents a matrix containing the corresponding simulated GEDI waveforms for the (presumably) same footprints.

d denotes the total number of features used for prediction.

5.3.2 Feature set

The predictor variables used to develop globally representative waveform-to-biomass models include key characteristics from GEDI waveforms and geographic data. In our study, we focus on features derived from a specific type of measurement within the waveform called relative height (RH). Incorporating geographic information has been shown to be crucial for developing globally representative models for AGBD prediction [150, 151]. Therefore, we include categorical features, such as world regions and dominant plant functional types (PFTs), in our prediction algorithms.

5.3.2.1 Relative heights

Relative height (RH) metrics provide information on how much energy is reflected back from vegetation at different heights above the ground. RHm refers to the height relative to ground elevation (in meters) at or below which m% of the total waveform energy, presumably from vegetation, has been received.

Deriving RH metrics is simpler for simulated GEDI waveforms compared to real GEDI data due to the absence of uncertainties in the ground-finding and the canopy top-finding process, which affects the calculation of RH metrics for real data [148]. Additionally, real GEDI waveforms are affected by inherent background light and electronic noise [48]. In this study, the RH metrics used for both real and simulated GEDI data are RHm with $m \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 98\}$.

5.3.2.2 Categorical features

Each sample pair in both the training and crossover datasets is linked to its corresponding geographic region and PFT classification. The PFT strata include deciduous broadleaf trees (DBT), evergreen broadleaf trees (EBT), evergreen needleleaf trees (ENT), and grasses, shrubs, and woodlands (GSW). The geographic regions are Africa, Australia-Oceania, Europe, North America, South America, and South Asia. Full details of the stratification process and the boundaries used are available in [46].

These categorical features are one-hot encoded in our analysis. For a categorical variable ordered from 1 to t, we assign $t-1$ binary features to our predictor variable, with the i'th ($1 \leq i \leq t-1$) feature being 1 if it belongs to the i'th class, and 0 otherwise. If all $t-1$ features are 0, the sample belongs to the tth class. Therefore, our GEDI waveforms are augmented with $(4-1)+(6-1)=8$ features, in addition to the ten RH metrics mentioned above. Thus, the total number of adopted features, d, is 18. Each unique combination of PFT and region is referred to as a stratum. Given five PFT categories and seven geographic regions, a total of 35 potential strata exist. However, due to data availability, six of these strata are empty in the crossover dataset, and eleven

are empty in the training dataset.

5.3.3 Methodology, using the co-Located crossover data

As described in Section 5.3.1, we have access to datasets comprising ALS-AGBD and real GEDI-ALS RH metrics, denoted as $(\mathbf{X'}_t, \mathbf{Y}_t)$ and $(\mathbf{X}_c, \mathbf{X'}_c)$, respectively. The ultimate goal, reflected in the GEDI04_A product, is to estimate AGBD from the real GEDI RH metrics. Currently, there is no comprehensive dataset of real GEDI waveforms paired with direct footprint-level measurements of AGBD for direct estimation purposes. To address this gap, we propose two approaches using the crossover dataset $(\mathbf{X}_c, \mathbf{X'}_c)$ to enhance the accuracy of AGBD estimation from real GEDI RH metrics. The sequential approach is detailed in Section 5.3.3.1, where we learn the model described in **(III)** from Section 5.2. The second approach, introduced in Section 5.3.3.2, focuses on the model described in **III-A** from Section 5.2.

5.3.3.1 Learning for model (III), sequential estimation

Let $X, X' \in \mathbb{R}^d$ denote the random row vectors of real GEDI and simulated GEDI (ALS) data, respectively, where d represents the number of adopted features. Let Y be the random variable representing GEDI footprint AGBD, which is the response variable here.

- **Conditional Independence Assumption (CIA)**: Our sequential approach relies on a critical assumption: that the real and simulated GEDI and ALS data in the crossover dataset are co-located and temporally aligned. In other words, each pair of real and simulated GEDI data in this dataset is assumed to correspond to the same footprint and forest structure. Real GEDI waveforms are inherently noisy, and deriving their RH metrics involves additional uncertainties due to ground-detection methods in real GEDI, which are not present in simulated GEDI [46]. Therefore, assuming no geo-location error, we further posit that the AGBD (Y) and the real GEDI waveform (X) are conditionally independent given the simulated GEDI waveform (X'), i.e., $p(Y|X, X') = p(Y|X')$. In simpler terms, given the

simulated GEDI RH matrix for a specific footprint, the value of the real GEDI waveform at the same footprint does not provide additional information about the AGBD at that location.

- **Mathematical justification**:

Lemma 15. *Let* $g(x') \triangleq \mathbb{E}[Y|X' = x']$*, and* $r(x) \triangleq \mathbb{E}[Y|X = x]$*. If* $\mathbb{E}[|Y|] < \infty$*, and* Y *and* X *are conditionally independent given* X'*, then* $r(X) = \mathbb{E}[g(X')|X]$*.*

Proof.

$$r(X) = \mathbb{E}[Y|X] \overset{(*)}{=} \mathbb{E}[\mathbb{E}[Y|X', X]|X] \overset{(**)}{=} \mathbb{E}[\mathbb{E}[Y|X']|X] = \mathbb{E}[g(X')|X] \quad (5.4)$$

Here, (**) directly results from the CIA. (*) follows from the tower property of conditional expectation. This property states that if random variable Y is defined on probability space $(\Omega, \mathbb{P}, \mathcal{F})$ with $\mathbb{E}[|Y|] < \infty$, and $\mathcal{H} \subseteq \mathcal{G} \subseteq \mathcal{F}$, we have

$$\mathbb{E}[\mathbb{E}[Y|\mathcal{G}]|\mathcal{H}] = \mathbb{E}[Y|\mathcal{H}]$$

(See [152, 153]). In our case, by defining $\mathcal{H} = \sigma(X)$ and $\mathcal{G} = \sigma(X, X')$, we can apply the tower property to Equation (*) in equation (5.4).

□

Remark 11. Lemma 15 demonstrates that any function capable of predicting $g(X')$ from X can also be used to predict Y from X. Consequently, rather than relying on the unavailable (X, Y) pairs, we can utilize $(X, g(X'))$ pairs. This principle underlies the sequential estimation process described below.

- **The sequential estimation process**:

1. Estimating $g(x')$: We first estimate $g(x') = \mathbb{E}[Y|X' = x']$ using the training dataset $(\mathbf{X}'_\mathbf{t}, \mathbf{Y_t})$. This estimator, denoted as $\widehat{g}(x')$, does not necessarily need to be a linear function. A variety of machine learning models can be employed for this estimation, including Linear Regression (LR), K-Nearest

Neighbors (KNN) [154], Random Forest (RF) [155], Extreme-gradient Boosting (XGB) [156], Artificial Neural Networks (ANN), and the previously established method by Kellner et al. [46].

2. Estimating $r(x)$: In the second stage, the crossover dataset is used to create a synthetic dataset of AGBD/real GEDI pairs. We form $(\widehat{g}(\mathbf{X}'_{\mathbf{c}}), \mathbf{X}_{\mathbf{c}})$, where $\widehat{g}(\mathbf{X}'_{\mathbf{c}})$ is a vector of size $N_c \times 1$ whose j'th entry is the value of $\widehat{g}$ applied to the j'th row of $\mathbf{X}'_{\mathbf{c}}$. Then, using this synthetic dataset, we estimate $\mathbb{E}[\widehat{g}(X')|X = x]$ and denote it by $\widehat{r}(x)$. This estimation can also be done through various model architectures, such as linear regression, random forest, k-nearest neighbors and extreme-gradient boosting as used in this paper.

Note that if (a) the estimation of $g(x') = \mathbb{E}[Y|X' = x']$ in the first stage is accurate (i.e., $\widehat{g}(x') \approx g(x') = \mathbb{E}[Y|X' = x']$), and (b) the estimation of $\mathbb{E}[\widehat{g}(X')|X = x]$ in the second stage is also accurate (i.e., $\widehat{r}(x) \approx \mathbb{E}[\widehat{g}(X')|X = x]$), then from equation (5.4), the final estimate $\widehat{r}(x)$ would closely approximate the true conditional expectation of AGBD given the real GEDI waveform (i.e., $\widehat{r}(x) \approx r(x)$). However, in practice, both stages involve estimation procedures that introduce errors. The accuracy of the final AGBD estimate $\widehat{r}(x)$ is ultimately influenced by the accumulation of errors from these estimation stages.

5.3.3.2 Specialized method for model III-A, single-stage estimation

Consider Model III-A (Equation (5.3)):

$$Y = g(X') + \epsilon'$$

$$X = X' + e$$

where ϵ' is independent of X' and has mean zero, and e is independent of X' and ϵ'. Under this model, expanding the squared error term when predicting AGBD using the same linear function on real GEDI data ($h(X)$) reveals two key components:

1. Mean Squared Error (MSE): measures the accuracy of AGBD prediction from the

training dataset.

2. Mean Squared Distance (MSD) or "consistency": measures the difference between AGBD predictions from real GEDI waveform and corresponding simulated GEDI in the crossover dataset.

$$\begin{aligned}\mathbb{E}\left[(h(X)-Y)^2\right] &= \mathbb{E}\left[\left((h(X')-Y)+(h(X)-h(X'))\right)^2\right] \\ &= \mathbb{E}\left[(h(X')-Y)^2+(h(X)-h(X'))^2\right] \\ &+ \mathbb{E}\left[2\,(h(X')-Y)\,(h(X)-h(X'))\right] \\ &\overset{(*)}{=} \mathbb{E}\left[(h(X')-Y)^2\right]+\mathbb{E}\left[(h(X)-h(X'))^2\right]\end{aligned} \tag{5.5}$$

(*) is because for a linear predictor function h we have:

$$\begin{aligned}\mathbb{E}\left[(h(X')-Y)\,(h(X)-h(X'))\right] &= \mathbb{E}\left[(h(X')-Y)\,h(e)\right] \\ &= \mathbb{E}\left[(h(X')-g(X')-\epsilon')\,h(e)\right] \\ &= \mathbb{E}\left[(h(X')-g(X'))\,h(e)\right]-\mathbb{E}\left[\epsilon' h(e)\right] \\ &= \mathbb{E}\left[h(X')-g(X')\right]\mathbb{E}\left[h(e)\right]-\mathbb{E}\left[\epsilon'\right]\mathbb{E}\left[h(e)\right] \\ &= \mathbb{E}\left[h(X')-g(X')\right]h(\mathbb{E}\left[e\right]) \\ &= 0\end{aligned} \tag{5.6}$$

The first component, MSE, is the traditional loss function used in the field, justified under assumptions (I) and (II) in Section 5.2. However, this derivation (5.5) under Model III-A motivates using the sum of MSE and MSD as a loss function in general cases, even when CIA fails.

Note that the vector X is always at the same spatio-temporal location as Y (by definition, as Y is the AGBD for the vegetation imaged by X). If X' is not at the same spatio-temporal location as Y, then X may carry information about Y not present in X', causing CIA to fail. In such cases, the justification for the sequential approach's second stage becomes questionable. Nevertheless, the estimator $h(x')$ from the first stage is still expected to be consistent across real and simulated GEDI data. Therefore,

applying the same prediction function $h(x')$ to real GEDI data (X) is expected to yield similar predictions, i.e., $h(X') \approx h(X)$. Thus, we can use the following loss function for single-stage estimation with consistency enforcement:

$$h^* = \arg\min_{h \in \mathcal{C}} \mathbb{E}\left[(Y - h(X'))^2\right] + \mathbb{E}\left[(h(X') - h(X))^2\right] \tag{5.7}$$

Here, $\mathcal{C}$ represents the class of functions to which the AGBD predictor is restricted.

If $\mathcal{C}$ is the class of linear functions, i.e., when $h(x') = x'\beta$ for $\beta \in \mathbb{R}^{d\times 1}$, the optimization problem in equation (5.7) has the solution $h^*(x') = x'\beta^*$, where β^* is given by:

$$\beta^* = \left(\mathbb{E}\left[X'^{\top}X' + (X' - X)^{\top}(X' - X)\right]\right)^{-1} \mathbb{E}\left[X'^{\top}Y\right] \tag{5.8}$$

Since the true underlying distributions are unknown, we replace the true expected values in equation (5.8) with the sample means from the training and crossover datasets. This yields:

$$\widehat{\beta} = \left(\mathbf{X'}_t^{\top}\mathbf{X'}_t + \frac{N_t}{N_c}(\mathbf{X'}_c - \mathbf{X}_c)^{\top}(\mathbf{X'}_c - \mathbf{X}_c)\right)^{-1} \left(\mathbf{X'}_t^{\top}\mathbf{Y}_t\right) \tag{5.9}$$

See the supplementary materials for details.

Remark 12. In the model II-A (5.1):

$$Y = X\gamma + \epsilon$$

$$X = X' + e$$

where e and ϵ have mean zero, ϵ is independent of X and (e, ϵ, X') are mutually independent. This model exhibits posterior mean equivalence, making the second term in Equation (5.7) unnecessary, and the crossover dataset would not be required for model training. However, the assumption that e is independent of ϵ is unrealistic.

To understand why, consider that e represents the inherent noise from on-orbit (real) waveforms. Since Y is not directly affected by these errors, the error term ϵ must account for all the real waveform-specific noise. Consequently, ϵ cannot be independent of e. Due to this, we based this section on Model III-A instead.

Remark 13. The second term in equation (5.7) can be interpreted as a regularization term. In the context of a linear model, it penalizes functions that show large discrepancies between predictions based on real and simulated GEDI data. This penalty is particularly strong for GEDI features where the difference between real and simulated values is substantial. While our supplementary material explores L_2 regularization (which penalizes all model coefficients equally), our experiments found it yielded equivalent results. This suggests that the noise in the GEDI data might be isotropic, meaning it affects all features similarly and no specific direction requires stronger penalization. The supplementary material also details other loss functions explored in this study.

5.3.4 Evaluation of performance of the models

5.3.4.1 Performance metrics

Equation (5.5) motivates the combined use of MSE and MSD as evaluation criteria. Integrating these two metrics ensures that our model not only performs well on the training data but also maintains consistency between the real and simulated GEDI data. This dual focus is crucial for achieving reliable and robust predictions in the presence of measurement error.

For any two functions h and f from $\mathbb{R}^d$ to $\mathbb{R}$, we define $MSE_h \triangleq \widehat{\mathbb{E}}\left[(Y - h(X'))^2\right]$ and $MSD_{h,f} \triangleq \widehat{\mathbb{E}}\left[(h(X') - f(X))^2\right]$. Here, $\widehat{\mathbb{E}}$ represents the sample expected value.

- **Sequential approach**: To evaluate the sequential model, we consider the following performance metric:

$$\sqrt{MSE_{\widehat{g}} + MSD_{\widehat{g},\widehat{r}}} \tag{5.10}$$

We also report each metric separately: $RMSE_{\widehat{g}} \triangleq \sqrt{MSE_{\widehat{g}}}$ and $RMSD_{\widehat{g},\widehat{r}} \triangleq \sqrt{MSD_{\widehat{g},\widehat{r}}}$.

Another performance metric we report is $RMSE_{\widehat{r}} \triangleq \sqrt{MSE_{\widehat{r}}}$.

- **single-stage approach**: To evaluate the second method (see Section 5.3.3.2), we use the following performance metric:

$$\sqrt{MSE_{\widehat{g}} + MSD_{\widehat{g},\widehat{g}}} \tag{5.11}$$

where $\widehat{g}(x) = x\widehat{\beta}$ as defined in equation (5.9). We also report each metric $RMSE_{\widehat{g}}$ and $RMSD_{\widehat{g},\widehat{g}}$ separately.

Remark 14. Under the Conditional Independence Assumption (CIA) that underpins the sequential method, the performance metric given in equation (5.10) becomes equivalent to the AGBD prediction error (measured by MSE) when using $\widehat{r}$ if $\widehat{g} = g$. In fact, the following lemma demonstrates that, under the CIA, the Mean Squared Error (MSE) of any estimator h on X (including $\widehat{r}$) is equivalent to the sum of MSE_g and $MSD_{g,h}$:

Lemma 16. *If Y and X are conditionally independent given X', then for $g(x') = \mathbb{E}[Y|X' = x']$, and any prediction function $h(x)$, we have*

$$\mathbb{E}[(Y - h(X))^2] = \mathbb{E}[(Y - g(X'))^2] + \mathbb{E}[(g(X') - h(X))^2] \tag{5.12}$$

Proof. Note that

$$\begin{aligned}
\mathbb{E}[(Y - h(X))^2] &= \mathbb{E}[(Y - g(X') + g(X') - h(X))^2] \\
&= \mathbb{E}[(Y - g(X'))^2] + \mathbb{E}[(g(X') - h(X))^2] \\
&\quad + 2\mathbb{E}[(Y - g(X'))(g(X') - h(X))] \\
&= MSE_g + MSD_{g,h}
\end{aligned} \tag{5.13}$$

The last equation holds because

$$\begin{aligned}\mathbb{E}\left[(Y - g(X'))\,(g(X') - h(X))\right] &= \mathbb{E}\left[\mathbb{E}\left[(Y - g(X'))\,(g(X') - h(X))\,|X'\right]\right] \\ &\overset{(*)}{=} \mathbb{E}\left[\mathbb{E}\left[Y - g(X')|X'\right]\mathbb{E}\left[g(X') - h(X)|X'\right]\right] \\ &= \mathbb{E}\left[(g(X') - g(X'))\,\mathbb{E}\left[g(X') - h(X)|X'\right]\right] \\ &= 0\end{aligned} \tag{5.14}$$

The equation in (*) is because, given X', the random variables $Y - g(X')$ and $g(X') - h(X)$ are conditionally independent. □

5.3.4.2 Cross validation

To assess the performance of Model 1, which involves two stages, we use a k-fold cross-validation, detailed as follows:

1. **Partitioning the Training Dataset**: The training dataset is randomly partitioned into k subsets. Each sample pair is assigned to one of the k parts uniformly at random.

2. **Partitioning the Crossover Dataset**: Similarly, the crossover dataset is randomly partitioned into k parts, with each sample pair uniformly assigned.

3. **Two-Stage Training**: For each $i \in 1, \ldots, k$:

   - The first stage (estimating $g(x')$) is conducted using all parts except the i'th part of the training dataset (simulated GEDI/AGBD). Let us denote it by $\widehat{g}_i$.

   - The second stage (estimating $r(x)$) is conducted using all parts except the i'th part of the crossover dataset. In other words, $\widehat{g}_i$ is applied on simulated GEDI from all parts except the i'th, and then the estimation of $r(x)$, denoted by $\widehat{r}_i$ is derived from estimation $\mathbb{E}[\widehat{g}_i(X')|X = x]$ on this training portion of crossover dataset.

   - After completing both stages, we predict AGBD using $\widehat{g}_i(X')$ and $\widehat{r}_i(X)$ on the i'th part of the training dataset (depending on the adopted performance

metric). We also find the predicted AGBD by $\widehat{g}_i(X')$ on the simulate GEDI on the i'th part of the crossover dataset, and $\widehat{r}_i(X)$ on the real GEDI of this the same part. (See the MSD term in evaluation metric in (5.10))

4. **Evaluation**: After completion of the previous steps, we have predicted AGBD values for every sample in the training dataset, each from its turn as the test part. Similarly, we have predicted AGBD values for every pair in the crossover dataset. This allows us to evaluate MSE_h (defined in Section 5.3.4.1) on every sample in the training dataset and $MSD_{h,f}$ on every sample in the crossover dataset.

The cross-validation process for the second model follows a similar procedure. The first and second steps are the same as described previously for model 1. Then here, for each $i \in \{1, \ldots, k\}$, the model is trained by optimizing the loss in equation (5.7), i.e., the first term in (5.7) uses the training part (all parts except the i'th) of the training dataset, and the second term in (5.7) uses the training part (all parts except the i'th) of the crossover dataset. Once the model is trained at each step i, it predicts AGBD on the i'th part of the training dataset and on both simulated GEDI and real GEDI data from the i'th part of the crossover dataset. These predicted values for every sample in both training and crossover datasets, all from $\widehat{g}(x) = x\widehat{\beta}$ are then used to evaluate the performance metric described in (5.11).

5.4 Related work

Our sequential approach shares conceptual similarities with studies in [157–159], particularly the Generalized Hierarchical Model-Based Estimation (GHMB) method proposed by Saarela et al [157]. However, their two-stage procedure is valid under the conditional independence assumption, which is not articulated in their paper. Moreover, GHMB relies solely on linear models estimated using generalized least squares (GLS) for both stages. In contrast, our proposed algorithm leverages the flexibility of pre-existing machine learning parametric or non-parametric models (e.g., random forests, k-nearest neighbor regression, extreme-gradient boosting, artificial neural networks)

during both stages. This allows us to potentially capture more complex relationships between features and AGBD compared to using GLS throughout. On the other hand, GHMB offers a more straightforward estimator of uncertainty.

5.5 Results

In the random partitioning phase of both datasets described in Section 5.3.4.2, we chose $k = 5$. For each $i \in \{1, 2, 3, 4, 5\}$, the training part of both datasets (all parts except the ith one) included between 66 to 90 percent of every non-empty stratum. This guarantees that for each i, both training and test datasets contain representatives of non-empty strata. We fixed this partition of both datasets and used it for every result shown below to facilitate comparison.

5.5.1 Model 1: Sequential approach

This section presents the results obtained from our sequential model (Model 1) as described in Section 5.3.3.1. We explore various machine learning paradigms at each stage to evaluate their effectiveness in predicting AGBD. We also include the Kellner23 algorithm for the first stage, as it was used to estimate $\widehat{g}(x')$.

For both stages, the algorightms of linear regression (LR), k-Nearest Neighbors (kNN), Random Forest (RF), and XGBoost (XGB) were trained using the Python libraries below respectively:

`sklearn.linear_model.LinearRegression`

`sklearn.neighbors.KNeighborsRegressor`

`sklearn.ensemble.RandomForestRegressor`

`xgboost.XGBRegressor`

Using KNN for both stages, we used 20 neighbors as the hyperparameter. For RF, we used 50 estimators and a maximum tree depth of 5. When using XGB, we set the

learning rate to 0.1 and the number of trees (`n_estimators`) to 20 for the first stage and 50 for the second.

For the Artificial Neural Network (ANN), we used a shallow ANN with an input layer, three hidden layers, and an output layer. The number of neurons in the first, second, and third layers were 32, 128, and 512, respectively. We set the maximum epochs to 500, used rectified linear unit (ReLU) nonlinearities, and set the kernel initializer as normal. The batch size used was 10, and the Python package employed was `keras.models.Sequential`.

The tables below summarize different performance metrics for various combinations of methods used in the first and second stages of the model. In Tables 5.1, 5.2, 5.3, and 5.4, each row represents the method used in the first stage, while each column is associated with the approach taken at the second stage.

Table 5.1: Performance metric $RMSE_{\widehat{g}}$ on training dataset for various ML techniques taken at the first stage

Method	$RMSE_{\widehat{g}}$
Kellner23	117.79
LR	114.45
KNN	117.22
RF	110.86
XGB	110.73
ANN	116.03

Note: Table 5.1 compares the performance of various machine learning algorithms when using the conventional approach in the field, which is to estimate $\mathbb{E}[Y|X]$ by using the dataset $(\mathbf{X'}_t, \mathbf{Y}_t)$. This approach assumes feature equivalence ($X = X'$) or posterior mean equivalence ($\mathbb{E}[Y|X = x] = \mathbb{E}[Y|X' = x]$) as discussed in Section 5.2.

However, if the underlying assumption of Model 1 (CIA) holds, then according to Remark 14, the appropriate performance metric to consider is $\sqrt{MSE_{\widehat{g}} + MSD_{\widehat{g},\widehat{r}}}$, which is shown in Table 5.3.

For comparison, the last column of Tables 5.2 and 5.3 includes the case where $\widehat{r} = \widehat{g}$, meaning the crossover dataset is not used, and AGBD is predicted solely with the

previously obtained $\widehat{g}$. This comparison helps explore the potential benefits of sequential estimation over the conventional approach.

Table 5.2: Performance metric $RMSD_{\widehat{g},\widehat{r}}$ on crossover dataset, for combination of various ML schemes taken for both stages. Each row represents the approach taken to estimate g, and each column is associated with the approach used for estimating r

	LR	KNN	RF	XGB	$\widehat{g}$
Kellner23	93.49	89.56	90.44	87.99	104.25
LR	71.83	69.10	72.69	67.25	76.21
KNN	56.53	52.34	54.65	51.68	58.64
RF	69.73	65.96	67.54	64.89	71.57
XGB	59.38	55.93	57.53	54.97	61.90
ANN	72.45	68.57	69.94	66.67	73.58

Table 5.3: Performance metric $\sqrt{MSE_{\widehat{g}} + MSD_{\widehat{g},\widehat{r}}}$ for combination of various ML models used at both stages

	LR	KNN	RF	XGB	$\widehat{g}$
Kellner23	150.38	147.97	148.51	147.02	157.30
LR	135.13	133.69	135.58	132.75	137.50
KNN	130.14	128.37	129.34	128.11	131.07
RF	130.97	129.00	129.82	128.46	131.96
XGB	125.65	124.06	124.79	123.63	126.86
ANN	136.79	134.78	135.48	133.82	137.39

Our results in Table 5.3 indicate that the other algorithms used in the first stage outperform the Kellner23 algorithm.

Furthermore, under the conditional independence assumption (CIA), since the performance metric of Table 5.3 represents the MSE of $\widehat{r}$ on real GEDI X (as shown in Lemma 16), comparing our sequential approach with the last column (which represents the case without crossover data) reveals that our sequential estimation approach with XGB for both stages significantly outperforms the Kellner algorithm without crossover data. However, it's important to note that the Kellner23 algorithm was specifically optimized for a different cross-validation procedure known as geographic cross-validation. This procedure is designed to enhance spatial generalization to domains outside the geographic location of the training data (see [46] for details). In contrast, our evaluation measures performance under regular cross-validation, as detailed in Section 5.3.4.2.

Additionally, the loss function employed by Kellner23 is distinct, incorporating unit transformations for both Aboveground Biomass Density (AGBD) and Relative Height (RH) metrics [46].

For each $\widehat{r}$ derived from the two stages from various combinations of models, we also report the RMSE for predicting AGBD on the training dataset, using the function $\widehat{r}$ in table 5.4:

Table 5.4: Performance metric $RMSE_{\widehat{r}}$

	LR	KNN	RF	XGB
Kellner23	135.53	127.52	123.09	124.69
LR	127.65	125.51	120.78	121.52
KNN	132.67	121.63	120.63	122.43
RF	140.67	124.96	122.81	121.54
XGB	141.83	125.77	126.96	124.82
ANN	136.20	130.85	125.59	123.55

A direct comparison between Tables 5.1 and 5.4 reveals that using $\widehat{g}$ on simulated GEDI waveforms typically yields slightly better performance than using $\widehat{r}$. This outcome is expected, as $\widehat{g}$ is derived from $g(x) = \mathbb{E}[Y|X' = x']$. However, it's important to note that $\widehat{r}$ is explicitly tailored to better generalize to real GEDI data under the Conditional Independence Assumption (CIA), as discussed in Remark 14. This emphasis on real-world applicability is crucial for making accurate predictions of AGBD.

5.5.2 Single-stage model

For the single-stage approach, we employed the linear estimator $\widehat{g}(x) = x\widehat{\beta}$, where $\widehat{\beta}$ is obtained by optimizing the loss function defined in equation (5.7). Table 5.5 presents the performance metrics for this model.

Table 5.5: Results for single-stage approach

$RMSE_{\widehat{g}}$	$RMSD_{\widehat{g},\widehat{g}}$	$\sqrt{MSE_{\widehat{g}} + MSD_{\widehat{g},\widehat{g}}}$
119.83	48.94	129.44

Table 5.5 indicates that single-stage approach outperforms the sequential estimation

procedure with linear estimators for both stages, as evidenced by a lower combined score ($129.44 < 135.13$), which is expected as the single-stage approach directly minimizes this performance metric.

While Table 5.5 presents the performance of the linear estimator obtained by optimizing the loss function in equation (5.7), it is also valuable to explore the potential benefits of non-linear estimators developed for Model 1 within the single-stage approach framework. Specifically, we evaluated the performance of the $\widehat{r}$ estimators developed in Model 1 using the metrics $RMSD_{\widehat{r},\widehat{r}}$ and $\sqrt{MSE_{\widehat{r}} + MSD_{\widehat{r},\widehat{r}}}$. Note that the $RMSE_{\widehat{r}}$ values for these estimators are already shown in Table 5.4.

Table 5.6: $RMSD_{\widehat{r},\widehat{r}}$

	LR	KNN	RF	XGB
Kellner23	48.98	62.98	57.57	53.08
LR	50.56	57.69	50.78	47.52
KNN	42.88	46.79	43.45	42.35
RF	42.59	55.03	48.77	44.05
XGB	37.33	47.43	41.93	38.85
ANN	43.35	55.84	46.42	44.09

Table 5.7: $\sqrt{MSE_{\widehat{r}} + MSD_{\widehat{r},\widehat{r}}}$

	LR	KNN	RF	XGB
Kellner23	144.10	142.22	135.88	135.51
LR	137.29	138.13	120.99	130.48
KNN	139.42	130.31	120.80	129.54
RF	140.82	125.17	123.00	121.72
XGB	146.66	134.41	127.12	130.72
ANN	136.35	131.06	125.77	123.72

5.6 Conclusion

We introduced models that either require or do not require crossover data for estimating aboveground biomass density (AGBD) from real GEDI waveform characteristics. Since models not needing crossover data have already been explored, we focused on those that do. We proposed two estimation procedures utilizing crossover data, one based

on a stronger assumption than the other, but both requiring co-location of the crossover data. Future research will determine the most suitable approach.

The results in the last column of Table 5.3 suggest that, under the conditional independence assumption, utilizing crossover data and sequential estimation with XGB for both stages can substantially enhance the accuracy of AGBD estimation from real GEDI using $\widehat{r}(x)$, outperforming the Kellner algorithm.

Moreover, although the single-stage approach offers the advantage of minimizing prediction error and ensuring consistency between real and simulated GEDI data, it is not yet clear whether there are specific situations where it would be preferable over the sequential approach. The single-stage model is justified by the linear model with additive noise as specified in equation (5.3) (assumption **III-A**), which is a specific instance of the Conditional Independence Assumption (CIA). The sequential approach, however, is already built on this CIA model. Therefore, no additional models beyond those covered by this CIA have been identified that would necessitate a preference for the single-stage method. Exploring this aspect could be a potential direction for future research.

5.7 Supplementary material

5.7.1 Derivation of equation (5.8)

Limited to the class of linear functions, equation (5.7) can be written as:

$$\beta^* = \arg\min_{\beta} \mathbb{E}\left[(Y - X'\beta)^2\right] + \mathbb{E}\left[((X' - X)\beta)^2\right] \tag{5.15}$$

To simplify, we expand the terms inside the expectations:

$$\begin{aligned}
\mathbb{E}\left[(Y - X'\beta)^2\right] + \mathbb{E}\left[((X' - X)\beta)^2\right] &= \mathbb{E}\left[(Y - X'\beta)^\top (Y - X'\beta)\right] + \\
&\quad \mathbb{E}\left[\beta^\top (X' - X)^\top (X' - X)\beta\right] \\
&= \mathbb{E}\left[Y^\top Y - Y^\top X'\beta - \beta^\top X'^\top Y + \beta^\top X'^\top X'\beta\right] \\
&\quad + \beta^\top \mathbb{E}\left[(X' - X)^\top (X' - X)\right]\beta \\
&= \mathbb{E}\left[Y^2\right] - 2\mathbb{E}\left[Y^\top X'\beta\right] + \beta^\top \mathbb{E}\left[X'^\top X'\right]\beta \\
&\quad + \beta^\top \mathbb{E}\left[(X' - X)^\top (X' - X)\right]\beta \\
&= \mathbb{E}\left[Y^2\right] - 2\mathbb{E}\left[YX'\right]\beta \\
&\quad + \beta^\top \mathbb{E}\left[X'^\top X' + (X' - X)^\top (X' - X)\right]\beta
\end{aligned}$$

Next, we take the gradient with respect to β and set it to zero to find the minimizer:

$$\begin{aligned}
&\frac{\partial}{\partial \beta}\left(\mathbb{E}\left[(Y - X'\beta)^2\right] + \mathbb{E}\left[((X' - X)\beta)^2\right]\right) \\
&= -2\mathbb{E}\left[X'^\top Y\right] + 2\mathbb{E}\left[X'^\top X' + (X' - X)^\top (X' - X)\right]\beta = \\
&= 0
\end{aligned}$$

Solving for β, we get:

$$\beta^* = \left(\mathbb{E}\left[X'^\top X' + (X' - X)^\top (X' - X)\right]\right)^{-1} \mathbb{E}\left[X'^\top Y\right]$$

Since Hessian $\mathbb{E}\left[X'^\top X' + (X' - X)^\top (X' - X)\right]$ is positive-definite, we have derived the estimator β^*.

5.7.2 More on assumption II-A

If the underlying models are given by equations (5.1), we have:

$$\begin{aligned} Y &= X\gamma + \epsilon \\ X &= X' + e \end{aligned} \tag{5.16}$$

where ϵ is independent of both X and X', and e is independent of X' and ϵ.

Note that

$$\mathbb{E}[Y|X'] = \mathbb{E}[X\gamma + \epsilon|X'] = \mathbb{E}[(X' + e)\gamma + \epsilon|X'] = X'\gamma + \mathbb{E}[e|X']\gamma + \mathbb{E}[\epsilon|X']$$

Since $\mathbb{E}[e|X'] = \mathbb{E}[e] = 0$ and $\mathbb{E}[\epsilon|X'] = \mathbb{E}[\epsilon] = 0$, we have:

$$\mathbb{E}[Y|X'] = X'\gamma$$

Remember that the ultimate goal is to estimate $\mathbb{E}[Y|X]$, which is equivalent to estimate γ under this model, as $\mathbb{E}[Y|X] = X\gamma$. Since we showed that $\mathbb{E}[Y|X'] = X'\gamma$, one can estimate γ by estimating $\mathbb{E}[Y|X']$ instead. One way to achieve this is through the ordinary linear estimator (OLS):

$$\gamma'^{*} = \left(\mathbb{E}[X'^{\top}X']\right)^{-1}\mathbb{E}[X'^{\top}Y]$$

Since $Y = X'\gamma + (e\gamma + \epsilon)$ and $e\gamma + \epsilon$ will have mean zero and is uncorrelated with X'. The error terms satisfy the requirements for least squares (zero mean, uncorrelated with the independent variable X'), making OLS a reliable method for obtaining a consistent estimator of Y.

In essence, the specific error structure allows us to leverage the readily available simulated GEDI data (X') to estimate the true relationship between the response variable (Y) and the real GEDI (X).

5.7.3 Other loss functions

When $\mathcal{C}$ is the class of linear functions in equation (5.7) in the main text, we consider other loss functions listed below.

1. In equation (5.7), one can consider penalizing the l_2-norm of the coefficients to prevent overfitting:

$$\widehat{\beta_\lambda^{(1)}} = \underset{\beta \in \mathbb{R}^{d\times 1}}{\arg\min}\, \mathbb{E}\left[(Y - X'\beta)^2\right] + \mathbb{E}\left[(X'\beta - X\beta)^2\right] + \lambda\|\beta\|_{NC}^2 \tag{5.17}$$

   Here, $\|\beta\|_{NC}^2 \triangleq \sum_{i\in S_{NC}} \beta_i^2$, where S_{NC} represents the set of features that are associated with the non-categorical features and not the offset. In more detail, these are the features that are not associated with binary variables, and also not the first coefficient β_1, which is associated with the offset.

2. Another loss function to consider is to optimize only the MSE on the training dataset $(\mathbf{X}'_t, \mathbf{Y}_t)$, adding the l_2 regularization to prevent overfitting:

$$\widehat{\beta_\lambda^{(2)}} = \underset{\beta \in \mathbb{R}^{d\times 1}}{\arg\min}\, \mathbb{E}\left[(Y - X'\beta)^2\right] + \lambda\|\beta\|_{NC}^2 \tag{5.18}$$

3. The next loss function we investigated aims to prevent severe extrapolation errors by ensuring that $g(X) = X\beta$ is not extreme relative to the typical prediction $g(X'_t)$ for the same categorical value.

$$\widehat{\beta_\lambda^{(3)}} = \underset{\beta \in \mathbb{R}^{d\times 1}}{\arg\min}\, \mathbb{E}\left[(Y - X'\beta)^2\right] + \lambda\mathbb{E}\left[\left(X\beta - \mu_{h(X)}\beta\right)^2\right] \tag{5.19}$$

   Here, $h(X)$ is the function that assigns the combination of categorical values to which X belongs, and μ_j is the estimated mean of the subset of $\mathbf{X}'_t$ that has the combination of categorical values associated with j.

4. The last loss to consider is as follows:

$$\widehat{\beta_\lambda^{(4)}} = \underset{\beta \in \mathbb{R}^{d\times 1}}{\arg\min}\, \mathbb{E}\left[(Y - X'\beta)^2\right] + \lambda\mathbb{E}\left[(X'\beta - X\beta)^2\right] \tag{5.20}$$

In equations (5.17), (5.18), (5.19), and (5.20), the hyperparameter λ can be determined by cross-validation on MSE+MSD. The steps are as follows:

(a) Data Partitioning: Randomly select a subset of both datasets $(\mathbf{X'}_t, \mathbf{Y}_t)$ and $(\mathbf{X}_c, \mathbf{X'}_c)$ for training purposes.

(b) Model Training: For different values of λ, find $\widehat{\beta_\lambda^{(i)}}$ using the training subsets of both datasets. Specifically, the training subset of the training dataset is used in the MSE term, and the training part of the crossover dataset is used in the MSD term in (5.17) and the second term in (5.19).

(c) Model Selection: Identify the λ that minimizes $\mathbb{E}\left[\left(Y - X'\widehat{\beta_\lambda^{(i)}}\right)^2\right] + \mathbb{E}\left[\left(X'\widehat{\beta_\lambda^{(i)}} - X\widehat{\beta_\lambda^{(i)}}\right)^2\right]$ on the held-out part of the datasets. The first term is evaluated on the held-out part of the training dataset, and the second term is evaluated on the held-out part of the crossover dataset. Denote this λ as λ^*. The selected model is now $\widehat{\beta_{\lambda^*}^{(i)}}$ for $i \in \{1, 2, 3, 4\}$.

Figure 5.1 illustrates the Root Mean Squared Error (RMSE) and the Root Mean Squared Distance (RMSD) for each of the specified loss functions across different λ values. These models were trained on the randomly selected training portions of both the training and crossover datasets and evaluated on the held-out test subsets. Note that $\widehat{\beta_1^{(4)}} = \widehat{\beta}$ in equation (5.8) in the main text. The plot also displays the minimum $\sqrt{MSE + MSD}$ along each curve. These results are included in the table below, along with the outcomes when the region indicator is incorporated as a predictor:

Table 5.8: Minimum $\sqrt{MSE + MSD}$ through different loss functions

Estimated Coefficients	$\sqrt{MSE + MSD}$	
	PFT augmented	PFT and region augmented
$\widehat{\beta}$	131.0915	127.9019
$\widehat{\beta^{(1)}_{\lambda^*}}$	130.9698	127.835
$\widehat{\beta^{(2)}_{\lambda^*}}$	130.9119	127.8116
$\widehat{\beta^{(3)}_{\lambda^*}}$	131.1461	128.8785
$\widehat{\beta^{(4)}_{\lambda^*}}$	131.0915	127.9019

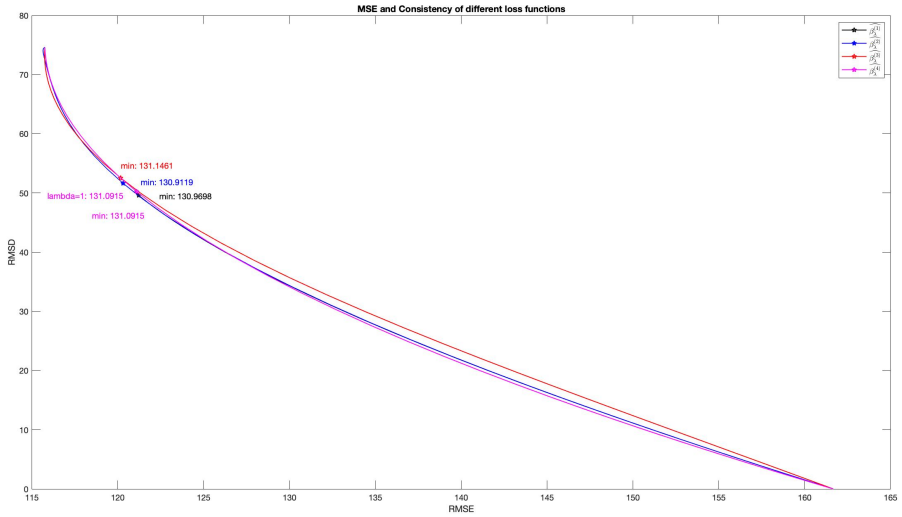


Figure 5.1: RMSE and RMSD for different loss functions for varying λ

The results show that the performance of the loss function with l_2 regularization (denoted by $\widehat{\beta^{(1)}_{\lambda^*}}$) is not significantly better than the baseline loss function without regularization (denoted by $\widehat{\beta}$). This suggests that l_2 regularization might not be necessary in this case.

There are two possible explanations for this observation:

- Large Dataset vs. Few Parameters: The dataset size (number of samples) is large compared to the number of parameters estimated in the model. In such cases, overfitting is less likely to be a major concern, and l_2 regularization, which aims to prevent overfitting by penalizing large parameter values, may not provide a significant benefit.

- Isotropic Noise: The similarity in performance between the $\widehat{\beta}$ and $\widehat{\beta^{(2)}_{\lambda^*}}$ suggests that the noise in the GEDI data may be isotropic, affecting all features uniformly, indicating that no particular direction necessitates stronger penalization.

Furthermore, as evident from the table, incorporating the region indicator as a predictor generally leads to a slight improvement in the minimum combined metric across all loss functions.

Bibliography

[1] Peng Zhang, Xingquan Zhu, and Yong Shi. Categorizing and mining concept drifting data streams. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 812–820, 2008.

[2] Geoffrey I Webb, Loong Kuan Lee, Bart Goethals, and François Petitjean. Analyzing concept drift and shift from sample data. *Data Mining and Knowledge Discovery*, 32:1179–1199, 2018.

[3] Peter Vorburger and Abraham Bernstein. Entropy-based concept shift detection. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 1113–1118. IEEE, 2006.

[4] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23:69–101, 1996.

[5] Indrė Žliobaitė and Jaakko Hollmen. Optimizing regression models for data streams with missing values. *Machine learning*, 99:47–73, 2015.

[6] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012.

[7] TASET SHIFT IN MACHINE LEARNING. Dataset shift in machine learning.

[8] Amos Storkey. When training and test sets are different: characterizing learning transfer. 2008.

[9] Dimitrios Michael Manias, Ali Chouman, and Abdallah Shami. Model drift in dynamic networks. *IEEE Communications Magazine*, 61(10):78–84, 2023.

[10] Tsam Kiu Pun, Mona Khoshnevis, Tommy Hosman, Guy H Wilson, Anastasia Kapitonava, Foram Kamdar, Jaimie M Henderson, John D Simeral, Carlos E Vargas-Irwin, Matthew T Harrison, et al. Measuring instability in chronic human intracortical neural recordings towards stable, long-term brain-computer interfaces. *bioRxiv*, pages 2024–02, 2024.

[11] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

[12] Marcos Salganicoff. Tolerating concept and sampling shift in lazy learning using prediction error context switching. *Lazy learning*, pages 133–155, 1997.

[13] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2018.

[14] Firas Bayram, Bestoun S Ahmed, and Andreas Kassler. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, 245:108632, 2022.

[15] Beata Jarosiewicz, Anish A Sarma, Daniel Bacher, Nicolas Y Masse, John D Simeral, Brittany Sorice, Erin M Oakley, Christine Blabe, Chethan Pandarinath, Vikash Gilja, et al. Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. *Science translational medicine*, 7(313):313ra179–313ra179, 2015.

[16] John E Downey, Nathaniel Schwed, Steven M Chase, Andrew B Schwartz, and Jennifer L Collinger. Intracortical recording stability in human brain–computer interface users. *Journal of neural engineering*, 15(4):046016, 2018.

[17] János A Perge, Mark L Homer, Wasim Q Malik, Sydney Cash, Emad Eskandar,

Gerhard Friehs, John P Donoghue, and Leigh R Hochberg. Intra-day signal instabilities affect decoding performance in an intracortical neural interface system. *Journal of neural engineering*, 10(3):036004, 2013.

[18] David Sussillo, Sergey D Stavisky, Jonathan C Kao, Stephen I Ryu, and Krishna V Shenoy. Making brain–machine interfaces robust to future neural variability. *Nature communications*, 7(1):13749, 2016.

[19] Alan D Degenhart, William E Bishop, Emily R Oby, Elizabeth C Tyler-Kabara, Steven M Chase, Aaron P Batista, and Byron M Yu. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. *Nature biomedical engineering*, 4(7):672–685, 2020.

[20] Paul Nuyujukian, Jonathan C Kao, Joline M Fan, Sergey D Stavisky, Stephen I Ryu, and Krishna V Shenoy. Performance sustaining intracortical neural prostheses. *Journal of neural engineering*, 11(6):066003, 2014.

[21] Cynthia A Chestek, Vikash Gilja, Paul Nuyujukian, Justin D Foster, Joline M Fan, Matthew T Kaufman, Mark M Churchland, Zuley Rivera-Alvidrez, John P Cunningham, Stephen I Ryu, et al. Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex. *Journal of neural engineering*, 8(4):045005, 2011.

[22] Gopal Santhanam, Michael D Linderman, Vikash Gilja, Afsheen Afshar, Stephen I Ryu, Teresa H Meng, and Krishna V Shenoy. Hermesb: a continuous neural recording system for freely behaving primates. *IEEE Transactions on Biomedical Engineering*, 54(11):2037–2050, 2007.

[23] Johan Wessberg and Miguel AL Nicolelis. Optimizing a linear algorithm for real-time robotic control using chronic cortical ensemble recordings in monkeys. *Journal of cognitive neuroscience*, 16(6):1022–1035, 2004.

[24] Sung-Phil Kim, Frank Wood, Matthew Fellows, John P Donoghue, and Michael J Black. Statistical analysis of the non-stationarity of neural population codes. In

The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006., pages 811–816. IEEE, 2006.

[25] Gopal Santhanam, Byron M Yu, Vikash Gilja, Stephen I Ryu, Afsheen Afshar, Maneesh Sahani, and Krishna V Shenoy. Factor-analysis methods for higher-performance neural prostheses. *Journal of neurophysiology*, 102(2):1315–1330, 2009.

[26] Wei Wu, Yun Gao, Elie Bienenstock, John P Donoghue, and Michael J Black. Bayesian population decoding of motor cortical activity using a kalman filter. *Neural computation*, 18(1):80–118, 2006.

[27] Sung-Phil Kim, John D Simeral, Leigh R Hochberg, John P Donoghue, Gerhard M Friehs, and Michael J Black. Point-and-click cursor control with an intracortical neural interface system by humans with tetraplegia. *IEEE transactions on neural systems and rehabilitation engineering*, 19(2):193–203, 2011.

[28] B Wodlinger, JE Downey, EC Tyler-Kabara, AB Schwartz, ML Boninger, and JL Collinger. Ten-dimensional anthropomorphic arm control in a human brain-machine interface: difficulties, solutions, and limitations. *Journal of neural engineering*, 12(1):016011, 2014.

[29] Jennifer L Collinger, Brian Wodlinger, John E Downey, Wei Wang, Elizabeth C Tyler-Kabara, Douglas J Weber, Angus JC McMorland, Meel Velliste, Michael L Boninger, and Andrew B Schwartz. High-performance neuroprosthetic control by an individual with tetraplegia. *The Lancet*, 381(9866):557–564, 2013.

[30] Wei Wang, Sherwin S Chan, Dustin A Heldman, and Daniel W Moran. Motor cortical representation of position and velocity during reaching. *Journal of neurophysiology*, 97(6):4258–4270, 2007.

[31] Tamraparni Dasu, Shankar Krishnan, Suresh Venkatasubramanian, and Ke Yi. An information-theoretic approach to detecting changes in multi-dimensional data

streams. In *Proc. Symposium on the Interface of Statistics, Computing Science, and Applications (Interface)*, 2006.

[32] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[33] Richard A Andersen, Tyson Aflalo, and Spencer Kellis. From thought to action: The brain–machine interface in posterior parietal cortex. *Proceedings of the National Academy of Sciences*, 116(52):26274–26279, 2019.

[34] Tyson Aflalo, Spencer Kellis, Christian Klaes, Brian Lee, Ying Shi, Kelsie Pejsa, Kathleen Shanfield, Stephanie Hayes-Jackson, Mindy Aisen, Christi Heck, et al. Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science*, 348(6237):906–910, 2015.

[35] Brian M Dekleva, Jeffrey M Weiss, Michael L Boninger, and Jennifer L Collinger. Generalizable cursor click decoding using grasp-related neural transients. *Journal of neural engineering*, 18(4):0460e9, 2021.

[36] Jeffrey M Weiss, Robert A Gaunt, Robert Franklin, Michael L Boninger, and Jennifer L Collinger. Demonstration of a portable intracortical brain-computer interface. *Brain-Computer Interfaces*, 6(4):106–117, 2019.

[37] Daniel Bacher, Beata Jarosiewicz, Nicolas Y Masse, Sergey D Stavisky, John D Simeral, Katherine Newell, Erin M Oakley, Sydney S Cash, Gerhard Friehs, and Leigh R Hochberg. Neural point-and-click communication by a person with incomplete locked-in syndrome. *Neurorehabilitation and neural repair*, 29(5):462–471, 2015.

[38] JD Simeral, Sung-Phil Kim, MJ Black, JP Donoghue, and LR Hochberg. Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array. *Journal of neural engineering*, 8(2):025027, 2011.

[39] Chethan Pandarinath, Paul Nuyujukian, Christine H Blabe, Brittany L Sorice, Jad

Saab, Francis R Willett, Leigh R Hochberg, Krishna V Shenoy, and Jaimie M Henderson. High performance communication by people with paralysis using an intracortical brain-computer interface. *elife*, 6:e18554, 2017.

[40] John E Downey, Jeffrey M Weiss, Sharlene N Flesher, Zachary C Thumser, Paul D Marasco, Michael L Boninger, Robert A Gaunt, and Jennifer L Collinger. Implicit grasp force representation in human motor cortical recordings. *Frontiers in neuroscience*, 12:801, 2018.

[41] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.

[42] Raymond J Carroll, David Ruppert, and Leonard A Stefanski. *Measurement error in nonlinear models*, volume 105. CRC press, 1995.

[43] Y Yi Grace. *Statistical analysis with measurement error or misclassification*. Springer, 2016.

[44] Laura Duncanson, John Armston, Mathias Disney, Valerio Avitabile, Nicolas Barbier, Kim Calders, Sarah Carter, Jérôme Chave, Martin Herold, Thomas W Crowther, et al. The importance of consistent global forest aboveground biomass product validation. *Surveys in geophysics*, 40:979–999, 2019.

[45] Ralph Dubayah, James Bryan Blair, Scott Goetz, Lola Fatoyinbo, Matthew Hansen, Sean Healey, Michelle Hofton, George Hurtt, James Kellner, Scott Luthcke, et al. The global ecosystem dynamics investigation: High-resolution laser ranging of the earth's forests and topography. *Science of remote sensing*, 1:100002, 2020.

[46] James R Kellner, John Armston, and Laura Duncanson. Algorithm theoretical basis document for gedi footprint aboveground biomass density. *Earth and Space Science*, 10(4):e2022EA002516, 2023.

[47] J Bryan Blair and Michelle A Hofton. Modeling laser altimeter return wave-

forms over complex vegetation using high-resolution elevation data. *Geophysical research letters*, 26(16):2509–2512, 1999.

[48] Steven Hancock, John Armston, Michelle Hofton, Xiaoli Sun, Hao Tang, Laura I Duncanson, James R Kellner, and Ralph Dubayah. The gedi simulator: A large-footprint waveform lidar simulator for calibration and validation of spaceborne missions. *Earth and Space Science*, 6(2):294–310, 2019.

[49] Mijail D Serruya, Nicholas G Hatsopoulos, Liam Paninski, Matthew R Fellows, and John P Donoghue. Instant neural control of a movement signal. *Nature*, 416(6877):141–142, 2002.

[50] John K Chapin, Karen A Moxon, Ronald S Markowitz, and Miguel AL Nicolelis. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature neuroscience*, 2(7):664–670, 1999.

[51] Johan Wessberg, Christopher R Stambaugh, Jerald D Kralik, Pamela D Beck, Mark Laubach, John K Chapin, Jung Kim, S James Biggs, Mandayam A Srinivasan, and Miguel AL Nicolelis. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 408(6810):361–365, 2000.

[52] Jose M Carmena, Mikhail A Lebedev, Roy E Crist, Joseph E O'Doherty, David M Santucci, Dragan F Dimitrov, Parag G Patil, Craig S Henriquez, and Miguel A L Nicolelis. Learning to control a brain–machine interface for reaching and grasping by primates. *PLoS biology*, 1(2):e42, 2003.

[53] Mikhail A Lebedev, Jose M Carmena, Joseph E O'Doherty, Miriam Zacksenhouse, Craig S Henriquez, Jose C Principe, and Miguel AL Nicolelis. Cortical ensemble adaptation to represent velocity of an artificial actuator controlled by a brain-machine interface. *Journal of Neuroscience*, 25(19):4681–4693, 2005.

[54] W Wu, A Shaikhouni, JR Donoghue, and MJ Black. Closed-loop neural control of cursor motion using a kalman filter. In *The 26th Annual International Conference*

of the IEEE Engineering in Medicine and Biology Society, volume 2, pages 4126–4129. IEEE, 2004.

[55] Leigh R Hochberg, Daniel Bacher, Beata Jarosiewicz, Nicolas Y Masse, John D Simeral, Joern Vogel, Sami Haddadin, Jie Liu, Sydney S Cash, Patrick Van Der Smagt, et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372–375, 2012.

[56] Leigh R Hochberg, Mijail D Serruya, Gerhard M Friehs, Jon A Mukand, Maryam Saleh, Abraham H Caplan, Almut Branner, David Chen, Richard D Penn, and John P Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099):164–171, 2006.

[57] Andrew B Schwartz, X Tracy Cui, Douglas J Weber, and Daniel W Moran. Brain-controlled interfaces: movement restoration with neural prosthetics. *Neuron*, 52(1):205–220, 2006.

[58] Shaomin Zhang, Yuxi Liao, Xiaoxiang Zheng, Weidong Chen, and Yiwen Wang. Decoding the nonstationary neural activity in motor cortex for brain machine interfaces. *International Journal of Imaging Systems and Technology*, 21(2):158–164, 2011.

[59] Yu Qi, Bin Liu, Yueming Wang, and Gang Pan. Dynamic ensemble modeling approach to nonstationary neural decoding in brain-computer interfaces. *Advances in neural information processing systems*, 32, 2019.

[60] Wei Wu and Nicholas G Hatsopoulos. Real-time decoding of nonstationary neural activity in motor cortex. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 16(3):213–222, 2008.

[61] Cynthia A Chestek, Aaron P Batista, Gopal Santhanam, M Yu Byron, Afsheen Afshar, John P Cunningham, Vikash Gilja, Stephen I Ryu, Mark M Churchland, and Krishna V Shenoy. Single-neuron stability during repeated reaching in macaque premotor cortex. *Journal of Neuroscience*, 27(40):10742–10750, 2007.

[62] Jose M Carmena, Mikhail A Lebedev, Craig S Henriquez, and Miguel AL Nicolelis. Stable ensemble performance with single-neuron variability during reaching movements in primates. *Journal of Neuroscience*, 25(46):10712–10716, 2005.

[63] Ian H Stevenson, Anil Cherian, Brian M London, Nicholas A Sachs, Eric Lindberg, Jacob Reimer, Marc W Slutzky, Nicholas G Hatsopoulos, Lee E Miller, and Konrad P Kording. Statistical assessment of the stability of neural movement representations. *Journal of neurophysiology*, 106(2):764–774, 2011.

[64] SI Helms Tillery, DM Taylor, and AB Schwartz. Training in cortical control of neuroprosthetic devices improves signal extraction from small neuronal ensembles. *Reviews in the Neurosciences*, 14(1-2):107–120, 2003.

[65] Matthew J McGinley, Martin Vinck, Jacob Reimer, Renata Batista-Brito, Edward Zagha, Cathryn R Cadwell, Andreas S Tolias, Jessica A Cardin, and David A McCormick. Waking state: rapid variations modulate neural and behavioral responses. *Neuron*, 87(6):1143–1161, 2015.

[66] Martin Vinck, Renata Batista-Brito, Ulf Knoblich, and Jessica A Cardin. Arousal and locomotion make distinct contributions to cortical activity patterns and visual encoding. *Neuron*, 86(3):740–754, 2015.

[67] Jay A Hennig, Emily R Oby, Matthew D Golub, Lindsay A Bahureksa, Patrick T Sadtler, Kristin M Quick, Stephen I Ryu, Elizabeth C Tyler-Kabara, Aaron P Batista, Steven M Chase, et al. Learning is shaped by abrupt changes in neural engagement. *Nature Neuroscience*, 24(5):727–736, 2021.

[68] Camillo Padoa-Schioppa, Chiang-Shan Ray Li, and Emilio Bizzi. Neuronal activity in the supplementary motor area of monkeys adapting to a new dynamic environment. *Journal of neurophysiology*, 91(1):449–473, 2004.

[69] Jerome N Sanes and John P Donoghue. Plasticity and primary motor cortex. *Annual review of neuroscience*, 23(1):393–415, 2000.

[70] James C Barrese, Naveen Rao, Kaivon Paroo, Corey Triebwasser, Carlos Vargas-Irwin, Lachlan Franquemont, and John P Donoghue. Failure mode analysis of silicon-based intracortical microelectrode arrays in non-human primates. *Journal of neural engineering*, 10(6):066014, 2013.

[71] Kevin Woeppel, Christopher Hughes, Angelica J Herrera, James R Eles, Elizabeth C Tyler-Kabara, Robert A Gaunt, Jennifer L Collinger, and Xinyan Tracy Cui. Explant analysis of utah electrode arrays implanted in human cortex for brain-computer-interfaces. *Frontiers in Bioengineering and Biotechnology*, 9:759711, 2021.

[72] Toshihisa Toyoda and Kazuhiro Ohtani. Testing equality between sets of coefficients after a preliminary test for equality of disturbance variances in two linear regressions. *Journal of Econometrics*, 31(1):67–80, 1986.

[73] Gregory C Chow. Tests of equality between sets of coefficients in two linear regressions. *Econometrica: Journal of the Econometric Society*, pages 591–605, 1960.

[74] Franklin M Fisher. Tests of equality between sets of coefficients in two linear regressions: An expository note. *Econometrica: Journal of the Econometric Society*, pages 361–366, 1970.

[75] Uri Rokni, Andrew G Richardson, Emilio Bizzi, and H Sebastian Seung. Motor learning with unstable neural representations. *Neuron*, 54(4):653–666, 2007.

[76] Sung-Phil Kim, John D Simeral, Leigh R Hochberg, John P Donoghue, and Michael J Black. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *Journal of neural engineering*, 5(4):455, 2008.

[77] Lek-Heng Lim, Rodolphe Sepulchre, and Ke Ye. Geometric distance between positive definite matrices of different dimensions. *IEEE Transactions on Information Theory*, 65(9):5401–5405, 2019.

[78] Arakaparampil M Mathai and Serge B Provost. Quadratic forms in random variables: theory and applications. *(No Title)*, 1992.

[79] Vikash Gilja, Chethan Pandarinath, Christine H Blabe, Paul Nuyujukian, John D Simeral, Anish A Sarma, Brittany L Sorice, János A Perge, Beata Jarosiewicz, Leigh R Hochberg, et al. Clinical translation of a high-performance neural prosthesis. *Nature medicine*, 21(10):1142–1145, 2015.

[80] Dawn M Taylor, Stephen I Helms Tillery, and Andrew B Schwartz. Direct cortical control of 3d neuroprosthetic devices. *science*, 296(5574):1829–1832, 2002.

[81] Richard A Andersen, Lawrence H Snyder, David C Bradley, and Jing Xing. Multimodal representation of space in the posterior parietal cortex and its use in planning movements. *Annual review of neuroscience*, 20(1):303–330, 1997.

[82] Mark L Homer, Arto V Nurmikko, John P Donoghue, and Leigh R Hochberg. Sensors and decoding for intracortical brain computer interfaces. *Annual review of biomedical engineering*, 15(1):383–405, 2013.

[83] David Achanccaray and Mitsuhiro Hayashibe. Decoding hand motor imagery tasks within the same limb from eeg signals using deep learning. *IEEE Transactions on Medical Robotics and Bionics*, 2(4):692–699, 2020.

[84] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

[85] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

[86] Simon Caton and Christian Haas. Fairness in machine learning: A survey. *ACM Computing Surveys*, 56(7):1–38, 2024.

[87] Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)*, 55(3):1–44, 2022.

[88] Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pages 120–129. PMLR, 2019.

[89] Chiappa Silvia, Jiang Ray, Stepleton Tom, Pacchiano Aldo, Jiang Heinrich, and Aslanides John. A general approach to fairness with optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3633–3640, 2020.

[90] Paula Gordaliza, Eustasio Del Barrio, Gamboa Fabrice, and Jean-Michel Loubes. Obtaining fairness using optimal transport theory. In *International conference on machine learning*, pages 2357–2365. PMLR, 2019.

[91] Ray Jiang, Aldo Pacchiano, Tom Stepleton, Heinrich Jiang, and Silvia Chiappa. Wasserstein fair classification. In *Uncertainty in artificial intelligence*, pages 862–872. PMLR, 2020.

[92] Luca Oneto, Michele Donini, and Massimiliano Pontil. General fair empirical risk minimization. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[93] Adrián Pérez-Suay, Valero Laparra, Gonzalo Mateo-García, Jordi Muñoz-Marí, Luis Gómez-Chova, and Gustau Camps-Valls. Fair kernel learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 339–355. Springer, 2017.

[94] Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Fair regression with wasserstein barycenters. *Advances in Neural Information Processing Systems*, 33:7321–7331, 2020.

[95] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34(12):5586–5609, 2021.

[96] Xiaolin Yang, Seyoung Kim, and Eric Xing. Heterogeneous multitask learning with joint sparsity constraints. *Advances in neural information processing systems*, 22, 2009.

[97] Kim-Han Thung and Chong-Yaw Wee. A brief review on multi-task learning. *Multimedia Tools and Applications*, 77(22):29705–29725, 2018.

[98] Morteza Alamgir, Moritz Grosse-Wentrup, and Yasemin Altun. Multitask learning for brain-computer interfaces. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 17–24. JMLR Workshop and Conference Proceedings, 2010.

[99] George Panagopoulos. Multi-task learning for commercial brain computer interfaces. In *2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 86–93. IEEE, 2017.

[100] Bernardino Romera Paredes, Andreas Argyriou, Nadia Berthouze, and Massimiliano Pontil. Exploiting unrelated tasks in multi-task learning. In *Artificial intelligence and statistics*, pages 951–959. PMLR, 2012.

[101] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine learning*, 73:243–272, 2008.

[102] Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

[103] Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000.

[104] Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, pages 567–580. Springer, 2003.

[105] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[106] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576, 2021.

[107] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.

[108] Tieliang Gong, Qian Zhao, Deyu Meng, and Zongben Xu. Why curriculum learning & self-paced learning work in big/noisy data: A theoretical perspective. *Big Data & Information Analytics*, 1(1):111–127, 2015.

[109] Yang Shu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Transferable curriculum for weakly-supervised domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4951–4958, 2019.

[110] Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. Curriculum learning for domain adaptation in neural machine translation. *arXiv preprint arXiv:1905.05816*, 2019.

[111] Georgios Zoumpourlis and Ioannis Patras. Motor imagery decoding using ensemble curriculum learning and collaborative training. In *2024 12th International Winter Conference on Brain-Computer Interface (BCI)*, pages 1–8. IEEE, 2024.

[112] Junpei Komiyama, Akiko Takeda, Junya Honda, and Hajime Shimao. Nonconvex optimization for regression with fairness constraints. In *International conference on machine learning*, pages 2737–2746. PMLR, 2018.

[113] Mike Brookes. The matrix reference manual. *Imperial College London*, 3:16, 2005.

[114] John Duchi. Properties of the trace and matrix derivatives. *Available electronically at https://web. stanford. edu/˜ jduchi/projects/matrix_prop. pdf*, 2007.

[115] Takeshi Amemiya. *Advanced econometrics*. Harvard university press, 1985.

[116] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

[117] Christopher M Bishop. Pattern recognition and machine learning. *Springer google schola*, 2:645–678, 2006.

[118] Esteve Corbera and Heike Schroeder. Governing and implementing redd+. *Environmental science & policy*, 14(2):89–99, 2011.

[119] Holly K Gibbs, Sandra Brown, John O Niles, and Jonathan A Foley. Monitoring and estimating tropical forest carbon stocks: making redd a reality. *Environmental research letters*, 2(4):045023, 2007.

[120] Scott J Goetz, Matthew Hansen, Richard A Houghton, Wayne Walker, Nadine Laporte, and Jonah Busch. Measurement and monitoring needs, capabilities and potential for addressing reduced emissions from deforestation and forest degradation under redd+. *Environmental Research Letters*, 10(12):123001, 2015.

[121] Edward TA Mitchard, Sassan S Saatchi, Alessandro Baccini, Gregory P Asner, Scott J Goetz, Nancy L Harris, and Sandra Brown. Uncertainty in the spatial distribution of tropical forest biomass: a comparison of pan-tropical maps. *Carbon balance and management*, 8:1–13, 2013.

[122] GC Hurtt, Stephen Wilson Pacala, Paul R Moorcroft, J Caspersen, E Shevliakova, RA Houghton, and B Moore Iii. Projecting the future of the us carbon sink. *Proceedings of the National Academy of Sciences*, 99(3):1389–1394, 2002.

[123] Michelle A Hofton, Jean-Bernard Minster, and J Bryan Blair. Decomposition of laser altimeter waveforms. *IEEE Transactions on geoscience and remote sensing*, 38(4):1989–1996, 2000.

[124] Hao Tang, Ralph Dubayah, Anu Swatantran, Michelle Hofton, Sage Sheldon, David B Clark, and Bryan Blair. Retrieval of vertical lai profiles over tropical rain forests using waveform lidar at la selva, costa rica. *Remote Sensing of Environment*, 124:242–250, 2012.

[125] Jason B Drake, Ralph O Dubayah, Robert G Knox, David B Clark, and J Bryan Blair. Sensitivity of large-footprint lidar to canopy structure and biomass in a neotropical rainforest. *Remote Sensing of Environment*, 81(2-3):378–392, 2002.

[126] Jérôme Chave, Maxime Réjou-Méchain, Alberto Búrquez, Emmanuel Chidumayo, Matthew S Colgan, Welington BC Delitti, Alvaro Duque, Tron Eid, Philip M Fearnside, Rosa C Goodman, et al. Improved allometric models to estimate the aboveground biomass of tropical trees. *Global change biology*, 20(10):3177–3190, 2014.

[127] Nicholas C Coops, Thomas Hilker, Michael A Wulder, Benoît St-Onge, Glenn Newnham, Anders Siggins, and JA Trofymow. Estimating canopy structure of douglas-fir forest stands from discrete-return lidar. *Trees*, 21:295–310, 2007.

[128] Laura I Duncanson, Ralph O Dubayah, and Brian J Enquist. Assessing the general patterns of forest structure: Quantifying tree and forest allometric scaling relationships in the u nited s tates. *Global Ecology and Biogeography*, 24(12):1465–1475, 2015.

[129] Erik Næsset, Terje Gobakken, Ole Martin Bollandsås, Timothy G Gregoire, Ross Nelson, and Göran Ståhl. Comparison of precision of biomass estimates in regional field sample surveys and airborne lidar-assisted surveys in hedmark county, norway. *Remote Sensing of Environment*, 130:108–120, 2013.

[130] Gregory P Asner and Joseph Mascaro. Mapping tropical forest carbon: Calibrating plot estimates to a simple lidar metric. *Remote Sensing of Environment*, 140:614–624, 2014.

[131] Richard M Lucas, AC Lee, and Peter J Bunting. Retrieving forest biomass through integration of casi and lidar data. *International journal of remote sensing*, 29(5):1553–1577, 2008.

[132] António Ferraz, Sassan Saatchi, Liang Xu, Stephen Hagen, Jerome Chave, Yifan Yu, Victoria Meyer, Mariano Garcia, Carlos Silva, Orbita Roswintiart, et al. Car-

bon storage potential in degraded forests of kalimantan, indonesia. *Environmental Research Letters*, 13(9):095001, 2018.

[133] David A Coomes, Michele Dalponte, Tommaso Jucker, Gregory P Asner, Lindsay F Banin, David FRP Burslem, Simon L Lewis, Reuben Nilus, Oliver L Phillips, Mui-How Phua, et al. Area-based vs tree-centric approaches to mapping forest carbon in southeast asian forests from airborne laser scanning data. *Remote Sensing of Environment*, 194:77–88, 2017.

[134] Liang Xu, Sassan S Saatchi, Aurélie Shapiro, Victoria Meyer, Antonio Ferraz, Yan Yang, Jean-Francois Bastin, Norman Banks, Pascal Boeckx, Hans Verbeeck, et al. Spatial distribution of carbon stored in forests of the democratic republic of congo. *Scientific Reports*, 7(1):15030, 2017.

[135] Laven Naidoo, Renaud Mathieu, Russell Main, Waldo Kleynhans, Konrad Wessels, Gregory Asner, and Brigitte Leblon. Savannah woody structure modelling and mapping using multi-frequency (x-, c-and l-band) synthetic aperture radar data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:234–250, 2015.

[136] Gaia Vaglio Laurin, Qi Chen, Jeremy A Lindsell, David A Coomes, Fabio Del Frate, Leila Guerriero, Francesco Pirotti, and Riccardo Valentini. Above ground biomass estimation in an african tropical forest with lidar and hyperspectral data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 89:49–58, 2014.

[137] Endre Hofstad Hansen, Terje Gobakken, Ole Martin Bollandsås, Eliakimu Zahabu, and Erik Næsset. Modeling aboveground biomass in dense tropical submontane rainforest using airborne laser scanner data. *Remote Sensing*, 7(1):788–807, 2015.

[138] Liviu Theodor Ene, Erik Næsset, Terje Gobakken, Ernest William Mauya, Ole Martin Bollandsås, Timothy G Gregoire, Göran Ståhl, and Eliakimu Zahabu. Large-scale estimation of aboveground biomass in miombo woodlands using airborne laser scanning and national forest inventory data. *Remote Sensing*

of Environment, 186:626–636, 2016.

[139] António Ferraz, Sassan Saatchi, Clément Mallet, Stéphane Jacquemoud, Gil Gonçalves, Carlos Alberto Silva, Paula Soares, Margarida Tomé, and Luisa Pereira. Airborne lidar estimation of aboveground forest biomass in the absence of field inventory. *Remote Sensing*, 8(8):653, 2016.

[140] Ana Hernando, Luis Puerto, Blas Mola-Yudego, José Antonio Manzanera, Antonio García-Abril, Matti Maltamo, and Rubén Valbuena. Estimation of forest biomass components using airborne lidar and multispectral sensors. *iForest-Biogeosciences and Forestry*, 12(2):207, 2019.

[141] Colin J Gleason and Jungho Im. Forest biomass estimation from airborne lidar data using machine learning approaches. *Remote Sensing of Environment*, 125:80–91, 2012.

[142] Erik Næsset and Terje Gobakken. Estimation of above-and below-ground biomass across regions of the boreal forest zone using airborne laser. *Remote Sensing of Environment*, 112(6):3079–3090, 2008.

[143] Hans-Erik Andersen, Jacob Strunk, Hailemariam Temesgen, Donald Atwood, and Ken Winterberger. Using multilevel remote sensing and ground data to estimate forest biomass resources in remote regions: A case study in the boreal forests of interior alaska. *Canadian Journal of Remote Sensing*, 37(6):596–611, 2011.

[144] Scott G Zolkos, Scott J Goetz, and Rdoi Dubayah. A meta-analysis of terrestrial aboveground biomass estimation using lidar remote sensing. *Remote sensing of environment*, 128:289–298, 2013.

[145] Hao Tang, Jason Stoker, Scott Luthcke, John Armston, Kyungtae Lee, Bryan Blair, and Michelle Hofton. Evaluating and mitigating the impact of systematic geolocation error on canopy height measurement performance of gedi. *Remote Sensing of Environment*, 291:113571, 2023.

[146] Sean P Healey, Zhiqiang Yang, Noel Gorelick, and Simon Ilyushchenko. Highly

local model calibration with a new gedi lidar asset on google earth engine reduces landsat forest height signal saturation. *Remote Sensing*, 12(17):2840, 2020.

[147] Victoria Meyer, Sassan Saatchi, António Ferraz, Liang Xu, Alvaro Duque, Mariano García, and Jérôme Chave. Forest degradation and biomass loss along the chocó region of colombia. *Carbon balance and management*, 14:1–15, 2019.

[148] Michelle Hofton, J Bryan Blair, Sarah Story, and Donghui Yi. Algorithm theoretical basis document (atbd). *Washington, DC, USA: NASA*, 2020.

[149] Laura Duncanson, James R Kellner, John Armston, Ralph Dubayah, David M Minor, Steven Hancock, Sean P Healey, Paul L Patterson, Svetlana Saarela, Suzanne Marselis, et al. Aboveground biomass density models for nasa's global ecosystem dynamics investigation (gedi) lidar mission. *Remote Sensing of Environment*, 270:112845, 2022.

[150] Ib Friis and Henrik Balslev. *Plant Diversity and Complexity Patterns: Local, Regional, and Global Dimensions: Proceedings of an International Symposium Held at the Royal Danish Academy of Sciences and Letters in Copenhagen, Denmark, 25-28 May, 2003*, volume 55. Kgl. Danske Videnskabernes Selskab, 2005.

[151] B Poulter, P Ciais, E Hodson, H Lischke, F Maignan, S Plummer, and NE Zimmermann. Plant functional type mapping for earth system models. *Geoscientific Model Development*, 4(4):993–1010, 2011.

[152] Rabindra Nath Bhattacharya and Edward C Waymire. *A basic course in probability theory*, volume 69. Springer, 2007.

[153] David Williams. *Probability with martingales*. Cambridge university press, 1991.

[154] Evelyn Fix. *Discriminatory analysis: nonparametric discrimination, consistency properties*, volume 1. USAF school of Aviation Medicine, 1985.

[155] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[156] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[157] Svetlana Saarela, Sören Holm, Sean P Healey, Hans-Erik Andersen, Hans Petersson, Wilmer Prentius, Paul L Patterson, Erik Næsset, Timothy G Gregoire, and Göran Ståhl. Generalized hierarchical model-based estimation for aboveground biomass assessment using gedi and landsat data. *Remote Sensing*, 10(11):1832, 2018.

[158] Svetlana Saarela, Sören Holm, Anton Grafström, Sebastian Schnell, Erik Næsset, Timothy G Gregoire, Ross F Nelson, and Göran Ståhl. Hierarchical model-based inference for forest inventory utilizing three sources of information. *Annals of Forest Science*, 73:895–910, 2016.

[159] Sören Holm, Ross Nelson, and Göran Ståhl. Hybrid three-phase estimators for large-area forest inventory using ground plots, airborne lidar, and space lidar. *Remote sensing of environment*, 197:85–97, 2017.