3-30-2023

Beyond Machine Learning: An fMRI Domain Adaptation Model for Multi-study Integration

Lauryn Michelle Burleigh
Louisiana State University and Agricultural and Mechanical College

BEYOND MACHINE LEARNING: AN FMRI DOMAIN ADAPTATION MODEL FOR MULTI-STUDY INTEGRATION

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Psychology

by
Lauryn Burleigh
B.S., University of New Orleans, 2014
M.A, Louisiana State University, 2019
May 2023

Table of Contents

Abstract

Traditional machine learning analyses are challenging with functional magnetic resonance imaging (fMRI) data, not only because of the amount of data that needs to be collected, adding a particular challenge for human fMRI research, but also due to the change in hypothesis being addressed with various analytical techniques. Domain adaptation is a type of transfer learning, a step beyond machine learning which allows for multiple related, but not identical, data to contribute to a model, can be beneficial to overcome the limitation of data needed but may address different hypothesis questions than anticipated given the analysis computation. This dissertation assesses a novel domain adaptation package, PyKale, created for cognitive fMRI data to determine the benefit and use it can have within cognitive research.

Introduction

Fear to Mental Imagery

Mental imagery, and the fear thereof, can largely impact a person from early in life, such as a child becoming afraid of a non-existent monster, throughout later life, such as adults associating uncontrollable life-or-death outcomes with imagined thoughts and prayers (Clasen, 2012; Muris et al., 2003; Singh, 2017). Mental imagery involves a perceptual-like experience without external input (Kosslyn et al., 2001). Importantly, many symptoms of anxiety and post-traumatic stress disorder (PTSD) are based in fear associated with mental imagery, for example intrusive memories and flashbacks (Arntz et al., 2007; Shin & Liberzon, 2010). Though clinical studies have found imagery related techniques for treating such disorders to be effective (Arntz et al., 2007; Holmes et al., 2007); (Grunert et al., 2007), an understanding of the representation and expression of fear to mental imagery has been lacking.

Mental images are depictions of stimuli constructed in the mind that cause sensory changes in the individual. They allow us to partake in mental events that aid in processes such as remembering, planning, navigating, and decision making (Pearson et al., 2015). Within the brain, mental images undergo processes similar to that of perception, for example, partaking in mental visual imagery processes interferes with visual perception (Horowitz, 1969). This is because mental images retain the sensory characteristics and neural processes of perceived stimuli, and utilize information of previously perceived stimuli to generate the image (Dadds et al., 1997; Kosslyn, 1988).

Recent research has begun using differential fear conditioning paradigms in order to better understand the relationship between mental imagery and fear (Mertens et al., 2020). In a standard visual perception differential fear conditioning paradigm, a visually presented conditioned stimulus (CS+) is paired with an aversive unconditioned stimulus (US), while a

second visual conditioned stimulus (CS-) is never paired with the US. As a result of the acquisition, the CS+ elicits a conditioned response (CR) of a larger magnitude than the CS-, also known as differential fear. The CR measured in human fear conditioning includes self-reported fear (LeDoux & Hofmann, 2018), skin conductance response (SCR), and brain activity measured using functional magnetic resonance imaging (fMRI). More recently, interest in how higher-order processes such as working memory (Carter et al., 1998), concepts (Dunsmoor et al., 2012; Grégoire & Greening, 2020), and mental imagery (Mertens et al., 2020) contribute to fear conditioning has grown

Fear Conditioning in the Theater of the Mind (FCTM) Study

In a recent meta-analysis of differential fear conditioning in fMRI, Fullana et al. observed a significant fear conditioning network that included the largest effects in the anterior insula (aIn) and dorsal anterior cingulate cortex (dACC) and dorsomedial prefrontal cortex (dmPFC) (Fullana et al., 2016). Regarding fear and mental imagery, Greening et al. (2021) observed generalization of differential fear reactivity to mentally imagining the stimuli when undergoing differential fear conditioning to visually perceiving the stimuli, as measured with SCR, self-reported fear, and right aIn activity (Greening et al., 2021). These results were later replicated and extended in two recent studies conducted by my collaborators and myself (Burleigh et al., 2022; Burleigh and Greening, 2023). The study which this proposal is built on, the Fear Conditioning in the Theater of the Mind (FCTM) fMRI experiment, sought to replicate and extend the work by Greening et al. (2021) and Burleigh et al. (2022), investigating the neural mechanisms of differential fear acquisition to visually perceived and mentally imagined stimuli (Burleigh and Greening, 2023).

The fMRI FCTM study investigated the acquisition and generalization of differential fear of and between visual perception and mental imagery through two phases. In the Visual Acquisition phase, differential fear conditioning occurred to visually perceived stimuli. The stimuli were also mentally imagined but were not directly conditioned to, therefore this phase investigated differential fear conditioning to visual perception and generalizing from visual perception to mental imagery to replicate Greening et al. (2021). In the Imagery Acquisition phase, the percept modalities were opposite and differential fear conditioning occurred to mentally imaging stimuli while the stimuli were also visually perceived but not directly conditioned to. The Imagery Acquisition phase extended Greening et al. (2021) by allowing for the investigation of differential fear conditioning to mental imagery and generalizing from mental imagery to visual perception (Burleigh and Greening, 2023).

The Visual Acquisition phase replicated Greening et al. (2021), finding significant activation in the right aIn not only during the differential fear acquisition of a visually perceived stimulus but also found significant activation in the right aIn during the generalization of differential fear acquisition from visual perception to mental imagery. The Imagery Acquisition phase extended Greening et al. (2021), finding a cluster of significant activation in the right aIn, frontal operculum, and inferior frontal gyrus during differential fear acquisition to mentally imagined stimuli. The right aIn was also significantly active during the generalization of differential fear acquisition from mental imagery to visual perception. When looking across both phases, there was also significant activation in the right aIn, frontal operculum, and inferior frontal gyrus in an overlap analysis of acquiring differential fear to visually perceived (i.e., Visual Acquisition phase) and acquiring differential fear to mentally imagined (i.e., Imagery

Acquisition phase) stimuli (Burleigh and Greening, 2023). All regions found to be significant in the FCTM study are part of the fear conditioning network defined by Fullana et al. (2016).

Amygdala in Fear

A potentially surprising note of the neural findings regarding differential fear in mental imagery is the lack of amygdala activation found (Greening et al., 2021; Burleigh and Greening, 2023), even though this area is highly associated with fear (Morrison & Salzman, 2010). A meta-analysis investigating the neural mechanisms of differential fear conditioning to physically present percepts has shown that while the aIn is commonly found, the amygdala is not (Fullana et al., 2016). This does not necessarily mean that the amygdala is not involved in fear conditioning, but it may indicate that the amygdala requires an increased temporal sensitivity.

The amygdala is a key neural component in fear acquisition in rodent models (Ono et al., 1995; Yin et al., 2018) and is particularly important during early fear acquisition in rodents (LaBar & LeDoux, 1996) and rhesus monkeys (Antoniadis et al., 2009) models. Yin et al. (2018) investigated this possibility using a trial-by-trial analysis in human participants and showed attenuation of amygdala activity throughout differential fear conditioning. Amygdala voxel patterns became less distinguished, as determined using a representational similarity analysis (Yin et al., 2018).

Further investigation of the amygdala in both physically perceived and mentally imagined differential fear acquisition, particularly in human participants, is needed. To investigate fear acquisition, a fear model can use a classifier defined to identify a CS+ versus a CS- on different trials. For example, Reddan et al. (2018) trained a classifier to distinguish CS+ versus CS- trials during a threat conditioning task, revealing a threat predictive network including the PFC, amygdala, periaqueductal gray, insula, globus pallidus, and dACC. This

whole-brain MVPA classifier was then used to investigate threat expression within neural activation maps during extinction, showing that both mentally imagined and physically present percepts contribute to extinction (Reddan et al., 2018). This study indicates that a classifier can be trained on the emotional association of a stimulus. Reddan et al. (2018) focused on fear extinction, therefore the investigation of the amygdala in fear acquisition is still needed. The study also used a leave-k-participants-out procedure, which does not allow neural activation across time to be investigated. A trial-wise procedure using a fear model to investigate the amygdala in the FCTM study would provide novel insight to the neural mechanisms of differential fear acquisition to visually perceived and mentally imagined stimuli, furthering greatly our understanding of disorders such as anxiety and PTSD.

Machine Learning Mechanisms

The time course of the amygdala was able to be investigated in human fear by Yin et al. (2017) due to the number of trials presented. In the trial-by-trial analysis, 120 trials were presented to each participant in a block design study. This large number of trials allows for greater power when able to reduce signal noise, however in order to complete so many trials, the inter-trial intervals (ITI) and length of time the stimuli are presented must be short (e.g., 1 sec. in Yin et al. (2017), resulting in an important limitation. When using short trials and ITI, the hemodynamic response function (HRF), the increase in blood flow proceeding neuronal activity, must be assumed to be linear (Soon et al., 2003). There is considerable variability in the features of the HRF across brain areas and individuals, however (D'Esposito et al., 2003; Handwerker et al., 2004; Kruggel & von Cramon, 1999). For example, Kruggel & von Cramon (1999) found variability of time until peak across voxels in a subject and Handwerker et al. (2004) found variability, both within subjects across regions and across subjects, in both the time to peak and

width of HRF. Shortened ITIs result in smaller responses, dramatically decreasing statistical efficiency (Dale, 1999), and therefore add undesired variance, which could not only impact detecting differences in responses but also highly increase the likelihood of Type II errors (Soon et al., 2003). In addition, conventional univariate analyses (i.e., general linear modeling; GLM) are unable to be performed on studies that use short trials and do not collect the full HRF. The need for a trial-by-trial investigation is of particular importance regarding the acquisition of fear condition to a visually perceived stimulus versus the acquisition of a mentally imagined stimulus. Mental imagery produces a weaker neural signal than physical perception (Ganis et al., 2004), however how this difference in neural signal between the percept modalities impacts and interacts with the acquisition of fear condition in each is unknown. Fear acquisition to mental imagery may take more trials to emerge, or the acquisition itself may be more variable during early acquisition.

Multi-Voxel Pattern Analysis

Needing a large number of trials is a big limitation with training complex machine learning models and computational analyses. Machine learning identifies the number of features (i.e., voxels) greatly outnumbering the observations (i.e., sample size) as the curse-of-dimensionality (Bellman, 1961). Primarily, the curse-of-dimensionality can mean that even if training items can be modeled in a large feature space, spurious correlations may solely be contributing to the solution which will not generalize. One method of addressing this is using ridge regression which can identify models with more features than examples if the signal is strong enough. This method uses a penalty on the optimization function for particularly large values. However, when the problem is complex, the signal is not incredibly strong, and aggressive feature reduction isn't ideal, even regularized models run on single subjects, such as

ridge regression, may not have enough evidence to determine features that are actually predictive of the target structure and are spuriously correlated.

As the amount of features increase, the amount of evidence needed to distinguish relevant signal from noise also increases. An additional, huge limitation, not only in cognitive computational neuroscience but in all machine learning domains, is the inability of traditional machine learning models to utilize previously gained knowledge or information. Generalization of knowledge is something humans are particularly adept at. For example, once trained on a melody, humans can generalize the knowledge gained on unfamiliar melodies with the new structure (Pfordresher et al., 2021). This generalization can also be unintentional such as generalizing the learned association of fear to a stimulus from one percept modality to another (i.e., physical perception to mental imagery and vice versa; Appendix A; Greening et al., 2021). A traditional machine learning computation, however, is isolated and doesn't transfer or generalize knowledge from other models or data sets.

The limitations discussed can be seen at play with the FCTM study in a Multivoxel Pattern Analysis (MVPA). MVPA is a computational machine learning analysis that learns the pattern of voxels in a region of interest for conditions. The analysis learns the pattern from the data used to train the model and attempts to identify which condition is being presented in the test data (Kaplan et al., 2015). Studies have successfully used MVPA to decode Gabor patch orientation gratings, the same stimuli used in FCTM, in V1-V3 of imagined stimuli (Albers et al., 2013). The FCTM study attempted to replicate this, training an MVPA model to discriminate the CS+ and CS- using all trials, both view and imagine, in the habituation task. The model was then tested using mentally imagined CS trials in the acquisition task in V1-V3. The model was unsuccessful in both the Visual Acquisition and Imagery Acquisition phases, producing

accuracies below 55% and $p > 0.05$. The model was also tested using visually perceived CS trials using the same parameters. It was only successful in discriminating the visually perceived CSs in the Imagery Acquisition phase, with an accuracy of only 52%, $p = 0.04$ (Burleigh and Greening, 2023). Visual perception in the FCTM study, given the minimal number of trials to train the model on, does not have enough power or data to discriminate CSs. The magnitude of activation when mentally imagining stimuli is of a much lesser magnitude than that of visually perceiving stimuli (Klein et al., 2000), resulting in an even greater need for additional data to train a model.

Traditional Machine Learning

Machine learning, a category of artificial intelligence, is a useful tool for multiple domains but can be particularly impactful for cognitive neuroscientists and functional magnetic resonance imagining (fMRI) researchers as it enables computers to learn a task based on data given, similar to how humans learn from experience, without explicitly programming the learning task. The process allows computers to modify their actions over multiple iterations, or attempts, to improve accuracy (i.e., number of times the action produces a correct result). The machine learning process is related to Bayesian statistics, a null hypothesis statistic testing alternative which allows prior knowledge to influence the test at hand, and computational statistics, which make statistical predictions via computers. Solving a problem via traditional machine learning involves a model (i.e., the task to learn or problem to solve) being trained on a dataset to learn the task question at hand, and being tested on subset of the original data (i.e., data of one run from each subject, or one subject, of the study used in training but not included in the training itself) (Alzubi et al., 2018). Traditional machine learning is that a dataset must include the same data for the learning task (i.e. data from only one study can be used in the model; see Figure 1).

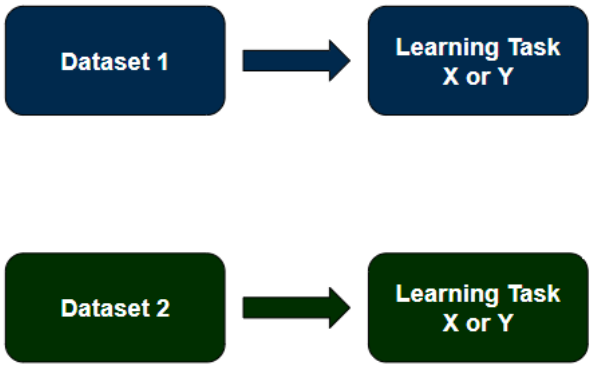


Figure 1. Traditional machine learning allows a dataset from a single study to contribute to the model's learning task.

Before solving the problem, the appropriate machine learning algorithm must be chosen by determining which one (or more) of the five questions data science can answer that the problem at hand fits into: classification problem (e.g., Is this A or B?), anomaly detection problem (e.g., Which is the odd one out?), regression problem (e.g., How much/many?), clustering problem (e.g., What is the structure/organization?), or a reinforcement learning problem (e.g., What should be done next?) (Alzubi et al., 2018). The majority of cognitive neuroscience machine learning algorithms are classification problems in which a fixed number of output classes are determined a priori, such as Yes/No, but are not required to be binary (Alzubi et al., 2018). Some common classification problems in cognitive neuroscience are fear conditioning conditioned stimulus classification (i.e., Is the trial a feared conditioned stimulus?), percept modality classification (i.e., Is the trial viewed or imagined/viewed or heard/etc.?), and category classification (i.e., Which animal/tool/etc. is present in the trial?). MVPA is a common fMRI traditional machine learning classification tool. An anomaly detection problem detects changes or anomalies (i.e., outliers) in a pattern (Alzubi et al., 2018). For example, in cognitive neuroscience it has been used to discriminate seafarers with altered default mode network

functioning from healthy controls (Shi et al., 2015). Regression problem algorithms handle continuous, numeric output, generally asking “How much?” or “How many?” (Alzubi et al., 2018). Cognitive fMRI researchers have used regression machine learning analyses for many purposes, from predicting ratings of features experienced in video games (Di Bono & Zorzi, 2008) to predicting alcohol misuse based on neuropsychological profiles (Whelan et al., 2014) to predicting math abilities and associated neural mechanisms following tutoring (Supekar et al., 2013). A clustering problem is one which creates clusters based on the similarity of the data structure. Due to the large number of features in fMRI data (i.e., many voxels), clustering algorithms are very commonly used to reduce the number of important features, such as independent components analysis (ICA) and principle component analysis (PCA) (Liang et al., 2016; Smolders et al., 2007), and are often used as a precursor in further machine learning algorithms (Alzubi et al., 2018). Finally, reinforcement problem algorithms learn the behavior/task to solve through trial and error in a continuously changing environment, using defined rewards and penalties (Alzubi et al., 2018). Though reinforcement algorithm models have not been widely used in cognitive neuroscience, one has been used to predict neural activation in various cortico-striatal loops during stimulus-action-reward association learning (Haruno & Kawato, 2006). Transfer learning can be used for all five questions, but this dissertation focuses on classification as it is the most common for cognitive neuroscience fMRI machine learning models.

Transfer Learning/Domain Adaptation

Transfer learning addresses the limitation of knowledge generalization and sharing in traditional machine learning models. Rather than allowing only identical data (i.e., one study) to contribute to a model, transfer learning can leverage information gained from related datasets,

allowing a larger and more diverse set of data to influence the model. This allows transfer learning to more similarly represent humans than traditional machine learning, as humans generalize knowledge learned to new situations both intentionally (Pfordresher et al., 2021) and unintentionally (Greening et al., 2021). Leverages information gained from related tasks, and subsequently domain adaptation, a specific transfer learning technique, can allow for a larger and more diverse selection of data to train a model (Zhou et al., 2019, 2020). Using this method, data from related but different tasks can be used to inform a model, increasing the accuracy by increasing the amount of data able to contribute to training the model (Zhou et al., 2019). Different datasets comprised on somewhat different examples may have overlapping content. Transfer learning provides ways of sharing information without treating all information input as identical, making the datasets more similar by, for example, excluding idiosyncratic signal and noise as domain adaptation does (Zhou, 2022).

Data input into transfer learning is either part of the source domain (i.e., the data to be leveraged) or target domain (i.e., the data to be classified). The task to be learned, which uses the source domain data, is the source task (blue 'Learning Task' box in Fig. 3a). Using the source domain data to train the source task results in the model gaining the knowledge of this learned source task. This knowledge-improved model then influences the target task which the target domain data is used for. When the same source and target domains are used (i.e., identical data from one study), and the same source and target tasks are used (i.e., one learning task), the analysis is equivalent to a traditional machine learning model.

There are three subtypes of transfer learning based on the relationship between the two domains and between the two tasks (see Table 1): (a) Inductive transfer learning uses the same domains and different but related tasks; (b) Unsupervised transfer learning uses different but

related domains and different but related tasks; and (c) *Transductive transfer learning* uses different but related domains and the same tasks. A domain has two components: a feature space (i.e., the space of all term vectors) and a marginal probability distribution of the learning sample, both of which are derived from the same dataset. When the source and target domain are different, it implies that either the source and target feature spaces, or the source and target probability distributions are not the same. Domain adaptation, the focus of this dissertation, specifies that the difference in domains occurs in the marginal probability distributions. Transductive transfer learning, and therefore domain adaptation, specifies that the learning tasks are the same, implying that a predictive function learned in the source domain can be adapted to use in the target domain if some unlabeled target-domain data is provided.

Table 1. The relationship between traditional machine learning and the tree transfer learning subtypes

Learning Type		Source + Target Domains	Source + Target Tasks
	Traditional Machine Learning	Same	Same
Transfer Learning	Transductive Transfer Learning [i.e., Domain Adaptation]	Different (but related)	Same
	Inductive Transfer Learning	Same	Different (but related)
	Unsupervised Transfer Learning	Different (but related)	Different (but related)

Before determining the types of transfer learning approaches that can and should be used for domain adaptation specifically, which situations transferring skills should be used and should not be used needs to be assessed. Domain adaptation should not be used if the source and target domains are not related. For a source and target domain to be related, an explicit or implicit relationship must exist between the feature spaces of the two domains (e.g., same study methodology at different locations, same study methods only altering stimuli, similar questions addressed in studies, etc.). A brute-force transfer with domains that are not related to each other

may be unsuccessful, and at worse, may hurt the learning performance in the target domain resulting in negative transfer. Negative transfer occurs when the source domain data and learning task lend to a reduced performance of learning in the target domain. Though little research has been done on this, negative transfer is a limitation of transfer learning, and therefore domain adaptation, to consider (Gui et al., 2017).

Aggregation of Subjects

The curse of dimensionality and the need for a large number of trials is based in the way fMRI data is processed in various analyses, particularly the aggregation of subjects. Not only does functional organization minorly vary across individuals, but additional noise unrelated to the hypothesis of interest can be introduced to the fMRI signal from the scan or brain itself, which can also vary. In univariate analyses, the meaningful signal is derived from a cluster of voxels, meaning they do not identify encoding in individual voxels but instead blur together groups of voxels emitting a signal. Therefore, inferences can be drawn by averaging activation across subjects since the voxel clusters are blurred. Machine and transfer learning techniques, including general domain adaptation, consider individual voxels, rather than blurring clusters of voxels. Considering individual voxels means machine learning analyses include the extra, unrelated noise to the signal, making it more difficult to differentiate a meaningful signal from noise because additional subjects are not included in averaging out the extra noise that can be conflated with the brain or scan dynamics of the particular subject. Leveraging and removing the covariance of the noise of the particular subject and the scan of the subject could aid in assessing other subjects to isolate the particular subject's noise profile, exposing voxel level signals that can be attributed to the scan in other subjects (figure 2).

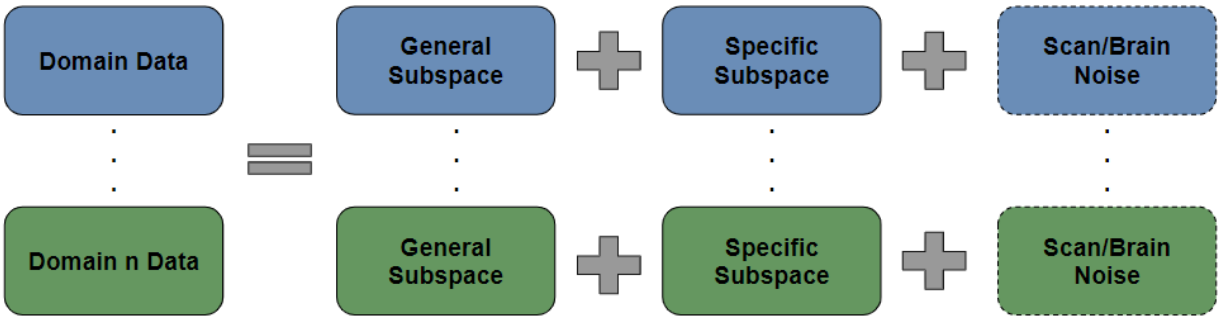


Figure 2. Each domain of fMRI data contains general subspace, specific subspace, and noise contributed by the scan and brain.

For example, participants retrieving the concept "dog" will evoke different distributed patterns of activity in their brains, the idiosyncratic specific subspace, partly due to each participant having had unique experiences with the word and creature being referred to, however, the visualization of a dog will activate occipitotemporal cortex in all participants, the general subspace. Each participant will also likely activate similar features such as "snout" and "tail". Additionally, noise unrelated to the hypothesis of interest can be introduced to the fMRI signal from the scan or brain itself, which can also vary. The process can be repeated to gather signals for additional stimuli, such as cat, which will have its own patterns of these three signals. Of particular note is that all participants have the same activation in the general subspace, however in the specific subspace, each participant shows the same activation in time one and time two of the dog, but the specific subspace is different across individuals (figure 3).

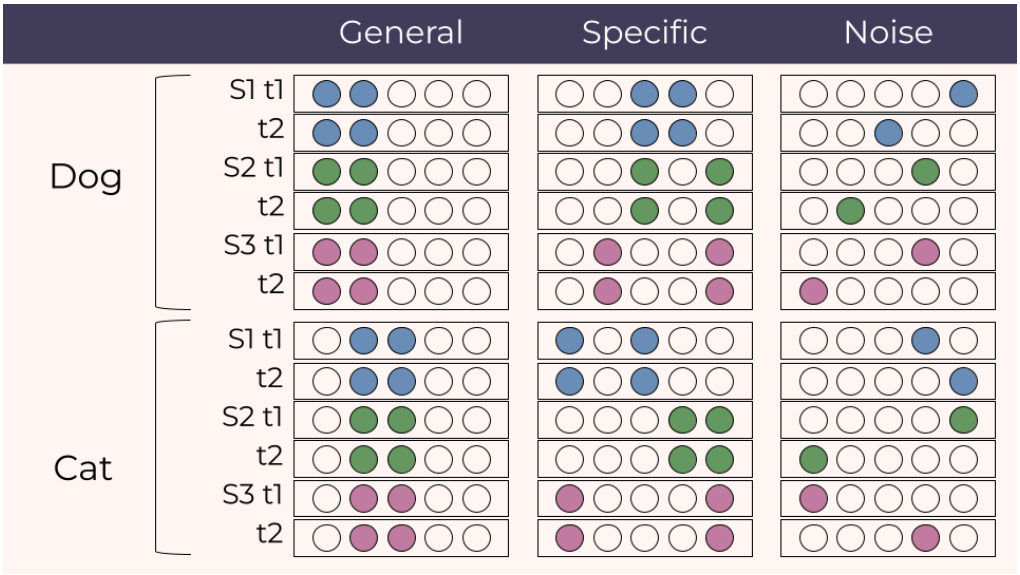


Figure 3. A representation of the neural activation by the general subspace, specific subspace, and noise for three subjects (S1, S2, and S3) each contributing two trials (t1, t2) for two stimuli (Dog, Cat). For each stimulus, the general subspace contains the same activation for each subject, the specific subspace contains the same activation at each trial for the specific subject, and the noise is sporadic for each trial of each subject.

Domain adaptation is a solution to minimal trials and the curse of dimensionality created by engineers and computer scientists in which the goal is to remove differences (e.g., brain and scan noise not related to brain activation) using all subjects' data by minimizing the difference of domains. The domain adaptation process makes the source and target domains as similar as possible and fit the model. This is not to be confused with classifier adaptation, in which a model of the source domain establishes priors that guide, but do not strictly constrain, the values that parameters might take in a model of the target domain (figure 4). Domain adaptation treats both the source and target domain from datasets that are related as one group to find and remove the shared subspace of noise, reducing the difference of the domains, then fitting a model to the unlabeled target domain data.

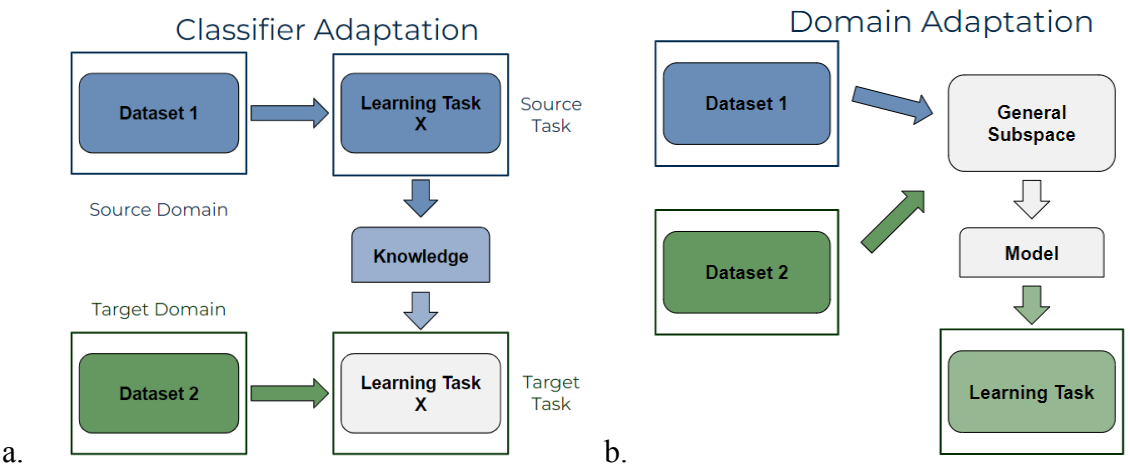


Figure 4. Classifier Adaptation versus Domain Adaptation. A. Classifier Adaptation uses the source domain to train a model, which then contributes knowledge to the learning task which the target domain information is fed into and classified using. B. Domain Adaptation uses both the source and target domain, reduces the difference between the domains, then uses the target task on the resulting model.

$\mathbf{CoIR_{LS}}$

Covariate-Independence Regularization ($CoIR_{LS}$) is a particularly fitting domain adaptation approach algorithm for cognitive fMRI researchers as it considers each dataset, as well as each subject, as a different domain, making it a multi-source domain adaptation technique. This means each unique experiment-subject combination is considered a distinct domain, allowing participants' individualistic brain activation to be taken into account in the final model. The CoIR algorithm considers a target dataset with labeled and unlabeled samples acquired from one or more source datasets, where the objective learning task is to predict the brain conditions of unlabeled target samples (Zhou, 2022).

Both the target and source experiments must have the same brain conditions to classify. To classify the conditions, the framework optimizes minimizing three aspects: 1) empirical error of labeled data, 2) dependence on domain covariates, and 3) model complexity (i.e., reducing the upper bound of generalization risk) (Zhou, 2022). The classifiers can be viewed as feature

mapping where samples of all domains are in the same distribution and project the input features to a one-dimensional output space, regularizing the hypothesis space to reduce model complexity. Additionally, the algorithm minimizes empirical error on labeled data and domain dependence on domain covariates further reducing model complexity. Minimalizing the prediction/empirical error and the domain dependence occurs in the framework simultaneously so can therefore be viewed as combining the benefits of domain-invariant classifier methods and domain dependence minimization mapping (Zhou, 2022).

The $CoIR_{LS}$ algorithm uses the regularized least squares classifier, a linear model which minimizes the sum of squared prediction errors, the sum of squared model parameters (e.g., the weight assigned to each voxel in an fMRI dataset) also known as the ridge penalty, and the Hilbert-Schmit Indepdence Criterion between the data and maxtrix of covariates which relates examples to subjects. Each unique experiment-subject combination is considered by encoding a matrix of experiment designs and subjects as domain covariates for domain adaptation, called one-hot encoding. Each subject is dummy-coded, allowing idiosyncratic variance to be found and removed, leaving a covariance structure that is shared across subjects. In this process, target domain samples are left unlabeled, which makes negative transfer highly unlikely. The model is fit over the source and target domain where neither domain is privileged, but the model is solely evaluated in the target domain. The output is a one-dimensional space of model weights for the classification task at hand, which can be used to identify the top 1% coefficients (in magnitude) to in turn visualize the brain areas of positive and negative activation for interpretation (Zhou, 2022).

Input

Inputs required include: 1) data matrix of source and target data; 2) vector of training labels; and 3) domain covariates.

Hyperparameters

Hyper-parameters (i.e., values that control the model's learning process) include: 1) alpha, which controls the constraint of coefficients; and 2) lambda which controls the importance of dependence on domain covariates. Optimal hyper-parameters can be found using leave-one-domain-out cross validation on source domain data.

***

There is a large need to focus on the general subspace of fMRI data in machine learning analyses, particularly when trying to study sensitively activating cognitive processes. Techniques are being developed in fields besides cognitive neuroscience, but work needs to be done to determine if, and how, useful they are. Though it does not solve all the problems of machine and transfer learning techniques, $CoIR_{LS}$ may bring cognitive neuroscience closer to being able to resolve some of the power and curse of dimensionality challenges commonly faced with analyses such as MVPA.

Methods

FCTM Study

Participants

The FCTM study enrolled, and results include, 31 healthy adults (17 females) between the ages of 18 and 33 (M=24.31 years). The sample size was determined based on previous psychophysiological and behavioral research using the same experimental paradigm (Burleigh et al., 2022) and based on previous research in fMRI comparing visual fear conditioning to imagery generalization (Greening et al., 2022). Participants were recruited through Louisiana State University's (LSU) human participant study pool and provided written informed consent. The experiments were approved by LSU's institutional review board. This study consisted of two phases: Visual Acquisition phase and Imagery Acquisition phase, which were counterbalanced across participants regarding which phase was presented first.

Materials

Gabor patches (sinusoidal gratings) were used as the conditioned stimuli throughout the study. In the Visual Acquisition phase, a rightward (45° angle with lines from lower left to upper right corner) and a horizontal (0° angle with lines from left to right) Gabor patch were used as the CS+ and CS-, counterbalanced across participants. For the Imagery Acquisition Phase leftward (135° angle, lines from lower right to upper left corner) and vertical (90° angle with lines from bottom to top) Gabor patches were used for the CSs, counterbalanced across participants. Thus, each of the two phases involved a distinct pair of conditioned stimuli. The CSs were presented within the participants' 8 degrees of visual angle. On each trial participants were provided an auditory instructional cue to either attend or imagine one of the Gabor patches. These trial-by-trial auditory instructions, "imagine right", "imagine horizontal", "attend right",

and “attend horizontal” in the Visual Acquisition phase, and “imagine left”, “imagine vertical”, “attend left” and “attend vertical” in the Imagery Acquisition phase, were delivered using Sensimetric MRI-compatible Insert Earphones and were produced using www.fromtexttospeech.com (Greening et al., 2021).

The US during the fear-conditioning trials was a mild electrical stimulation (administered using the STMISOC and STM100C modules of BIOPAC, Goleta, CA, USA). Two electrodes were placed on the fingertips of the first and second fingers of the non-dominant hand. The intensity of shock was individually set by each participant at a level that was “uncomfortable but not painful” (M= 4.43 mA), as consistent with previous research (Greening et al., 2016; Knight et al., 2005).

Procedure and Design

Once in the MRI scanner, participants completed two distinct phases: the Visual Acquisition phase and the Imagery Acquisition phase. The phases were counterbalanced across participants regarding which was presented first. Each phase began with 6 habituation runs followed by 6 fear acquisition runs. The purpose of the habituation runs was to familiarize participants with the task including the imagery and it allowed habituating participants initial responses to the auditory cues and the Gabor patches (Jiang et al., 2021). The participants also completed the Likert-style questionnaire after each phase (Burleigh and Greening, 2023).

The overall trial structure was identical for both the Visual Acquisition and Imagery Acquisition phases (figure 5) and was adapted from recent psychophysiological research from the lab (Burleigh et al., 2022).

Imagery Acquisition Phase

The Imagery Acquisition phase began with 6 habituation runs, each of which consisted of 8 trials. Each run contained two CS+ imagine without shock trials, two CS- imagine trials, two CS+ view trials, and two CS- view trials. The trial order was fully randomized.

Next, participants completed 6 fear acquisition runs, each of which included 12 trials. Each run contained two CS+ imagine with shock trials, two CS+ imagine without shock trials, four CS- imagine trials, two CS+ view trials, and two CS- view trials. Thus, imagining CS+ was paired with the US 50% of the time and viewing the CSs was never paired with shock in this phase. The trial order was pseudo-random (Greening et al., 2022). Each run began and ended with a CS- imagine trial. The first run in this phase also included a CS+ imagine with shock trial as the second trial to begin the fear association early. The second CS+ imagine with shock trial was randomly presented within the second half of the run. For the remaining 3 runs, the first CS+ imagine trial was randomly presented in the first half of the run and the second CS+ imagine trial was randomly presented in the second half of the run. Trials including a shock are not used in analyses so as to avoid any potential US confound in our modeling of CS+ trials. We also excluded the first and last CS- imagine trials of each run. This is the same approach we used in Greening et al. (2022) and allowed us to avoid the potentially confounding orienting effect generated on the first trial of a run and allow us to ensure that the number of trials in each first level regressor was equal across conditions of interest for the analysis. All other trials not otherwise noted were randomized within the run.

Visual Acquisition Phase

The Visual Acquisition phase was similar to the Imagery Acquisition phase in structure, as it included 6 habituations runs followed by 6 fear acquisition runs, but with several notable

exceptions. First, a novel pair of conditioned stimuli were used. Second, each fear acquisition run included two CS+ view with shock trials, two CS+ view without shock trials, four CS- view trials, two CS+ imagine trials, and two CS- imagine trials. In other words, only when viewing the CS+ did participants receive a shock (50% reinforcement rate), and no shock was ever delivered during imagery of the CSs.

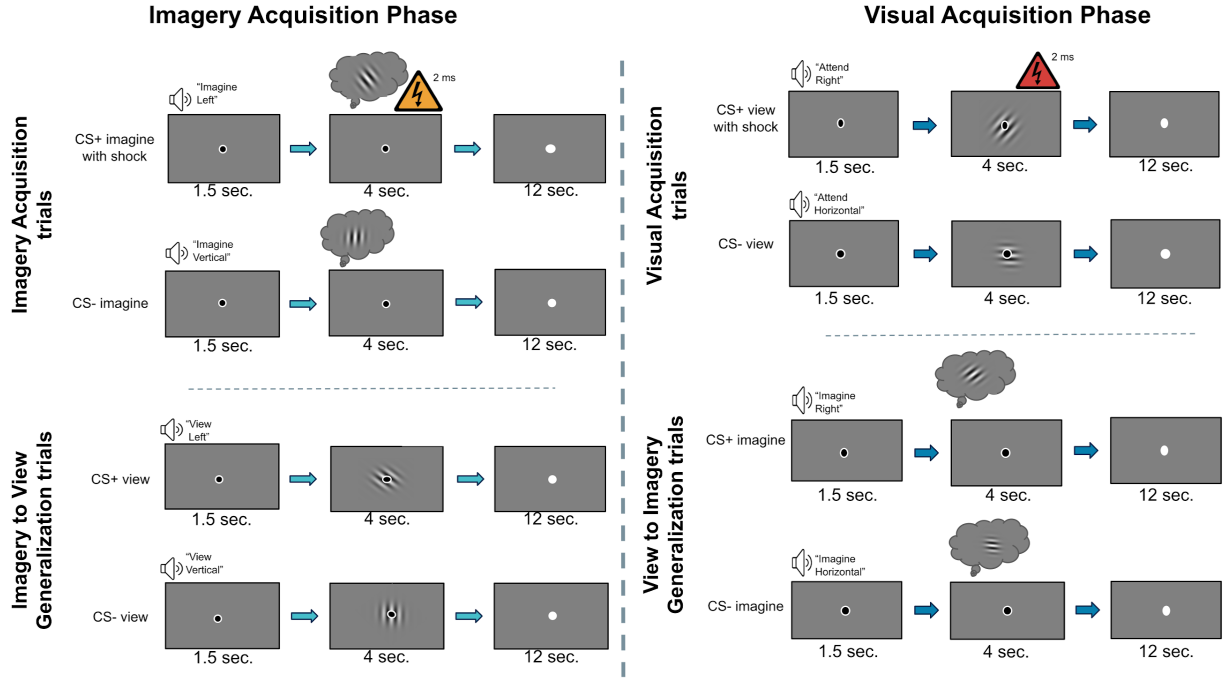


Figure 5. Trial structure. Participants imagined and viewed the patches associated with each of the Visual Acquisition and Imagery Acquisition phases, respectively.
Top left: A CS+ view trial during Visual Acquisition with 2 ms mild electrical stimulation that co-terminates with the Gabor Patch. Bottom left: A CS- view trial during Visual Acquisition. Top right: A CS+ imagine trial during Imagery Acquisition with 2 ms mild electrical stimulation that co-terminates with the 4 second imagery epoch. Bottom right: A CS- imagine trial during Imagery Acquisition.

Image Acquisition

Subjects were scanned using a 3T GE DISCOVERY MR750w MRI scanner with a 32-channel head coil. Each participant first received a high resolution, T1 weighted, anatomical scan covering the whole brain (repetition time=8.78ms; echo time=3.79ms; FOV=22.4cm; voxel

size=3.5 mm isovoxels; 64X64 matrix). The following procedures were then followed for each phase.

Twenty-four fMRI scans (6 habituation and 6 acquisition for each phase) were run to measure BOLD changes. The fMRI images were taken with a T2*-gradient echo-planar imaging sequence (TR=2000ms; TE=25ms; FOV=22.4 cm; 64X64 matrix). Complete brain coverage was obtained with 76 slices of 3.5 X 3.5 mm in plane with a slice thickness of 3.5 mm, forming 3.5 X 3.5 X 3.5 mm voxels. Each functional run began with collecting 3 dummy volumes to account for T1 equilibrium effects, discarded as part of the preprocessing steps of data analysis. The total number of volumes per run, including dummy volumes, varied according to task type, but not phase: habituation = 73 volumes, acquisition = 109 volumes.

fMRI Preprocessing

Both individual and group analyses were processed using FEAT (FMRI Expert Analysis Tool) in FSL (FMRIB's Software Library, www.fmrib.ox.ac.uk/fsl) version 5.0.9. Registration of functional images to the high resolution (T1-weighted) structural image and the standard space image was performed using FLIRT (Jenkinson et al., 2002; Jenkinson & Smith, 2001). The pre-statistics processing applied were as followed: motion correction using MCFLIRT (Jenkinson et al., 2002); slice-timing correction using Fourier-space time-series phase-shifting; non-brain removal using BET (Smith, 2002); spatial smoothing using a Gaussian kernel of FWHM 7 mm; grand-mean intensity normalization of the entire 4D dataset by a single multiplicative factor; high-pass temporal filtering (Gaussian-weighted least-squares straight line fitting, with sigma = 50.0s). The time-series statistical analysis was carried out using FILM with local autocorrelation correction (Woolrich, Ripley, Brady, & Smith, 2001).

MVPA

Multivariate pattern analysis (MVPA) was used to find a working, easy CS classification problem using the FCTM fMRI data outlined above. The model was trained on either view trials from the Visual Acquisition phase task or imagery trials from the Imagery Acquisition phase task, and tested whether the model could accurately identify whether a CS+ or CS- trial was presented in one run of held out testing data of the phase used to train the model. The MVPAs were carried out using a support vector machine (SVM) classifier in PyMVPA (Hanke et al., 2009) on each individual participant using a leave-one-run-out validation. As is the default in PyMVPA, the SVM hyperparameter 'C', which controls the width of the margin and number of support vectors to balance potential training misclassifications, was determined via automatic scaling according to the norm of the data. A binary whole brain mask was derived from the group alignment to standard MNI space, as all participant specific data used for the MVPAs were registered to MNI space. Preprocessing and first-level modeling of the fMRI data used for MVPA include those listed above. Each trial was modeled independently in the first-level GLM, including smoothing with a 7mm FWHM kernel, which has been shown previously to improve classifier performance (Hendriks et al., 2017).

None of the trials in which shock was presented were used. First the classifier was trained on view CS+ versus view CS- trials during 5 of the 6 Visual Acquisition phase task runs. To estimate the performance of the classifier model, we used cross-validation to use each run as the held-out test data.

Algorithms

$CoIR_{LS}$ Analysis

In this step, a leave-k-subjects-out-style paradigm using the FCTM data was used. In this case, one subject was considered the target domain (n = 1) and all other subjects (n = 30) were used as the source domain (see Appendix A or GitHub for code: https://github.com/lburleigh/lb_dissertation; see Appendix B or link for Jupyter Book Guide to $CoIR_{LS}$: https://lburleigh.github.io/da-guide/). All source domain data and 1 of the 6 runs of the target domain subject were labeled. The $CoIR_{LS}$ algorithm was dual cross-validated, repeating the classification with each subject being reserved for use as the target domain, mimicking the easy and hard problems of steps one and two, and repeating the classification with each run of each target domain subject being labeled. As determined in Easy MVPA, the CS domain adaptation focused on classifying the view CS+ versus the view CS- trials of the Visual Acquisition phase task and classifying the imagine CS+ versus the imagine CS- trials of the Imagery Acquisition phase task. The model was trained using the 5 unlabeled target subject domain runs and tested using the 1 labeled target subject domain run.

The same pre-processed data that was used in the MVPA was used in $CoIR_{LS}$, including the same MNI whole-brain mask. Additionally, due to $CoIR_{LS}$ using a 1:1 comparison of voxels which correlate with labels across participants, voxels that produced 0 activation across all runs were removed (1900 voxels) and the remaining voxels were z-scored to the training data. To determine the ideal alpha and lambda hyper-parameters, a hyper-parameter search was performed, running potential values for each (1e-2 to 1e+9) on the respective phase task for the respective trials of interest (Zhou, 2022; figure 6). The optimal hyperparameter for each subject was identified and used for the respective model, with the subject hyperparameter being used

when said subject was used as the target domain (see Appendix C for Visual Acquisition subject plots). The result of CoIRLS is a one-dimensional space of model weights which were then used in subsequent analyses below.

Ridge Regression

The ridge regression analysis followed the same procedures as the CoIRLS with the exception of participants being included, resulting in two ridge regressions run. One ridge used only one participant data in the analysis while another used all participants, mimicking $CoIR_{LS}$. Ridge regression is a type of linear regression used in machine learning used to reduce the complexity of models by having a penalty term applied to the cost function, making this method a good comparison to $CoIR_{LS}$ that is not a form of domain adaptation. While ridge regression has the penalty function lambda applied, it does not contain alpha as a hyperparameter.

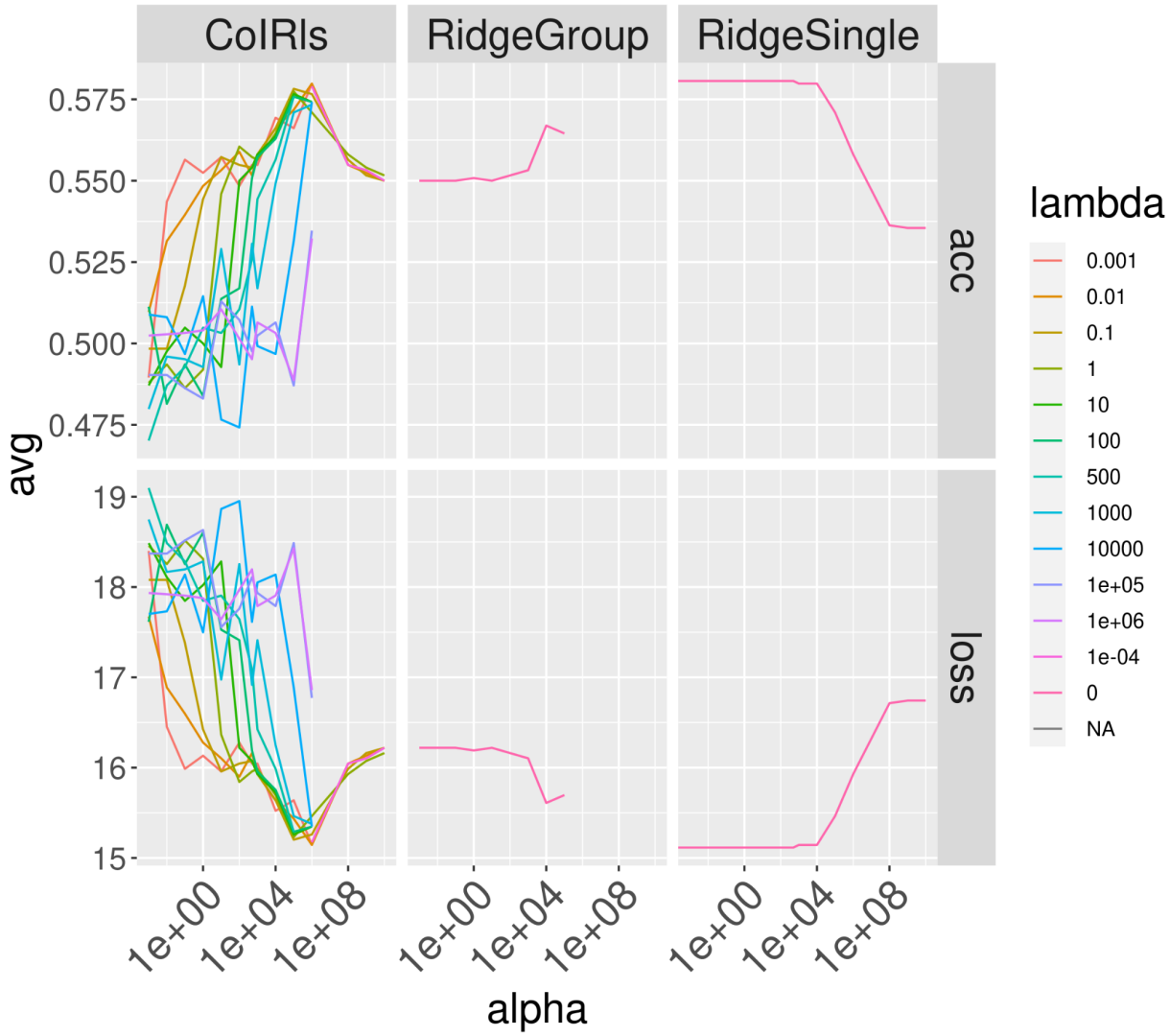


Figure 6 - This line plot shows the hyper-parameters searched in the Visual Acquisition task for view trials. Ridge Group refers to the ridge regression in which all participants are included while Ridge Single refers to the ridge in which only one participant was run.

Results

MVPA

"Easy" MVPA

Testing the classifier on view CS+ versus view CS- trials from the Visual Acquisition phase task produced a group-level accuracy of 0.58 (SD = 0.17; Figure 6 - blue Visual Acquisition bar). Each classifier was assessed using a one-sample, one-tailed t-test of the accuracy received from each participant compared to chance (0.50). A one-tailed test was used because an accuracy of lower than chance is meaningless. The one-sample, one-tailed t-test for classifying viewed CSs in the Visual Acquisition phase using all runs was statistically significant, $t(30) = 2.68$, $p = 0.005$. Next, the classifier was trained on imagine CS+ versus imagine CS- trials during 5 of the 6 Imagery Acquisition phase task runs, again using a cross-validation for the held-out test run. Testing the classifier on imagine CS+ versus imagine CS- trials from the Imagery Acquisition phase task, the MVPA produced a group-level accuracy of 0.59 (SD = 0.14; Figure 6 - blue Imagery Acquisition bar), which also showed significance in a t-test compared to chance, $t(30) = 3.53$, $p = 0.0007$.

"Hard" MVPA

Once the easy problem of step one was identified (i.e., CS classification in the acquisition task of each phase), step two involved removing data from the easy problem to 'break' the model turning the easy problem into a hard problem. The problem was deemed hard and sufficiently 'broken' when classification accuracy no longer produced significance. To attempt to break the model, the easy problem was repeated for each phase with the exception of the training dataset reduced to three runs, rather than five runs, and again tested the model on one hold-out run. The model was given the first four runs of the phase, one of which was again used for each hold-out

run cross-validation. This proved to be unsuccessful at breaking the model as both phases continued to produce significant classification accuracy: Visual Acquisition phase CS classification accuracy was 0.58 (SD = 0.16; Figure 7 - yellow Visual Acquisition bar), $t(30) = 2.79$, $p = 0.005$ and the Imagery Acquisition phase classification accuracy was 0.57 (SD = 0.14; Figure 6 - yellow Imagery Acquisition bar), $t(30) = 2.71$, $p = 0.006$.

Due to the significant classification found when performing leave-one-run-out cross validation on runs 1 through 4, the model training and test was repeated for each phase using only runs one and two (i.e., runs 3 and 4, in addition to runs 5 and 6, were excluded). Reducing the training dataset to one run was successful in breaking the model: Visual Acquisition phase CS classification accuracy was 0.56 (SD = 0.19; Figure 7 - grey Visual Acquisition bar), $t(30) = 1.67$, $p = 0.052$ and the Imagery Acquisition phase classification accuracy was 0.55 (SD = 0.22; grey Imagery Acquisition bar), $t(30) = 1.32$, $p = 0.098$. The hard problem was therefore deemed to be identical to the easy problem, but with the dataset reduced to only the first two runs of each phase contributing to running the model. Making the problem hard allows for step three, implementing transfer learning, to show that transfer learning can provide support enough to the machine learning process that changes to results and different understanding of the data are identified.

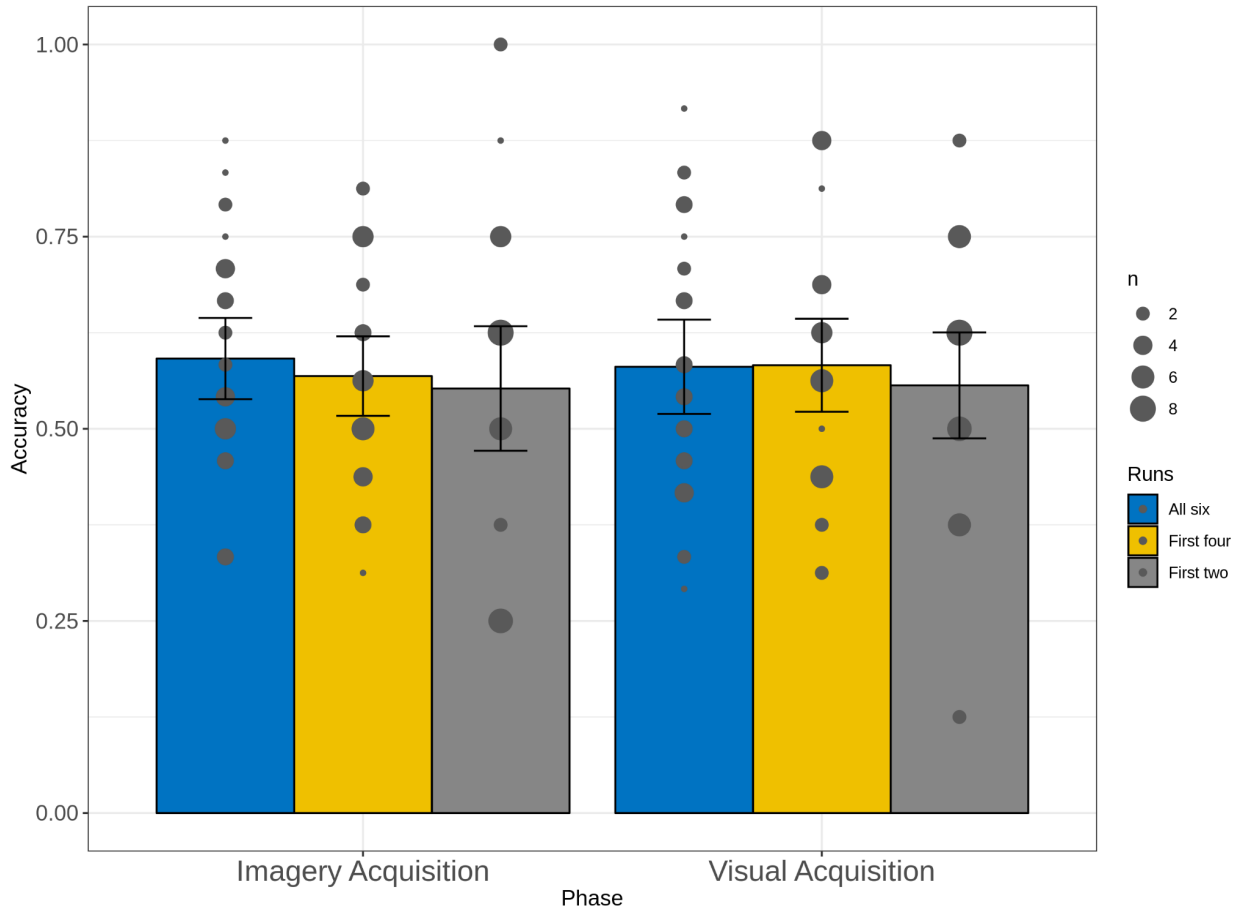


Figure 7. Bar graph of the classification accuracy results from Steps 1 and 2, with bar height representing mean classification accuracy of all participations. Bars are separated with the three Imagery Acquisition run selections on the left and three Visual Acquisition run numbers on the rightEach bar is colored according to the number of runs with far left blue bar for the model in which all runs were included, the center yellow bar for the model with the first four runs, and the right grey bar for the model using only the first two runs. Error bars indicate confidence intervals and data points are plotted with size of the point indicating the number of participants receiving each classification accuracy.

$CoIR_{LS}$ and Ridge Regression

Two repeated measure ANOVAs, one for each phase, were run on the average accuracy classification values received during testing for each subject, where each ANOVA model factor included three levels: $CoIR_{LS}$, ridge regression with all subjects (Ridge Group), and ridge regression with one subject (Ridge Single). The means and standard deviations of each model for

each phase are reported in Table 2. Neither ANOVA (tables 3 and 4) revealed significant effects, indicating there was no difference in average subject test accuracy for any of the three models (figure 8).

Table 2. The mean and standard deviation of the average test accuracy of each subject for each model run for the Imagery Acquisition phase and Visual Acquisition phase data.

	Imagery Acquisition Mean	Imagery Acquisition SD	Visual Acquisition Mean	Visual Acquisition SD
$CoIR_{LS}$	0.5564516	0.1433513	0.5796371	0.1070959
Ridge Group	0.5645161	0.1655654	0.5826613	0.1109896
Ridge Single	0.6048387	0.1414504	0.5745968	0.1158733

Table 3. Repeated Measure ANOVA results for the Imagery Acquisition phase, testing average subject test accuracy difference of the three models.

	df	Sum Sq	Mean Sq	F value	p-value
Model	2	0.0417	0.02083	2.007	0.143
Subjects	30	1.416	0.04721		
Error	60	0.6227	0.01038		

Table 4. Repeated Measure ANOVA results for the Visual Acquisition phase, testing average subject test accuracy difference of the three models.

	df	Sum Sq	Mean Sq	F value	p-value
Model	2	0.00103	0.0005145	0.184	0.832
Subjects	30	0.9487	0.03162		
Error	60	0.16774	0.002796		

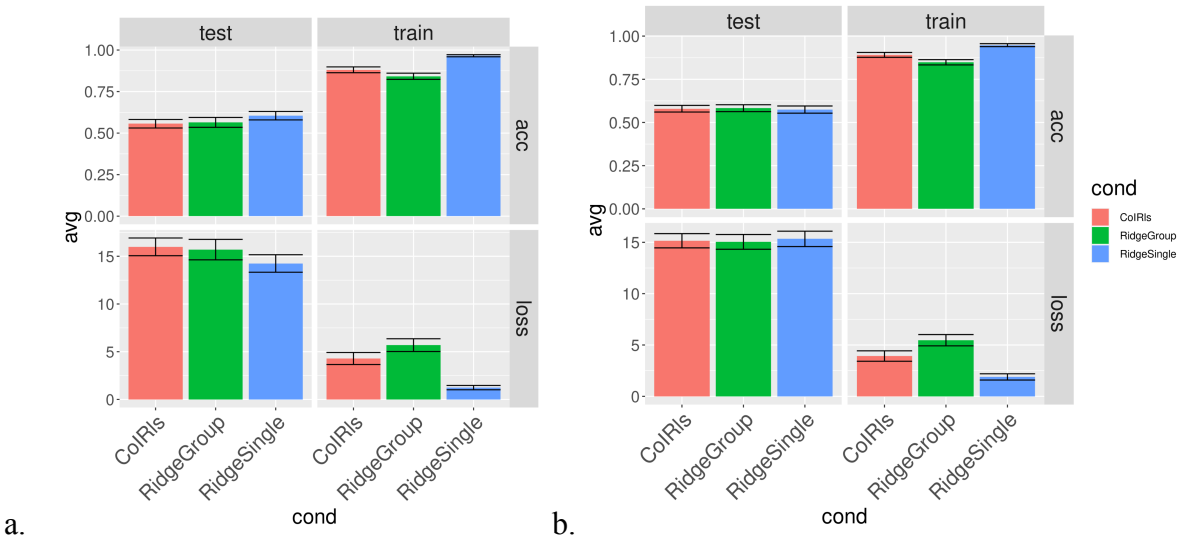


Figure 8. Bar plot and standard error bars of the average reported for each model (pink - $CoIR_{LS}$, green - Ridge False, blue - Ridge True) where the first column is values received during testing and second column is values during training. The top row of each plot is the accuracy and bottom row of each plot is loss. Error bars represent the standard error. A. Imagery Acquisition phase, b. Visual Acquisition phase

Conclusion

The curse of dimensionality is a major concern and limitation in progressing the analysis of fMRI data. Machine learning techniques, such as MVPA, allow classification to be performed on fMRI data but the need for a large number of trials and data to be input into the model to have enough power for the model to perform means this technique is not accessible or possible for many researchers. Not only is fMRI data collection expensive and limiting in the amount of time a participant can contribute, but the cognitive process being investigated often does not lend to the collection of many trials. In an fMRI data set in which a single experiment is performed by multiple participants, the cognitive process manifests in neural activity that is, in part, idiosyncratic to each participant and in part to neural activity that is similar across them. For example, two participants retrieving the concept "dog" will evoke different distributed patterns of activity in their brains, partly due to each participant having had unique experiences with the word and creature being referred to, however, the visualization of a dog will activate occipitotemporal cortex in both participants. Additionally, each participant will likely activate similar features such as "snout" and "tail". Domain adaptation refers to a class of machine learning techniques designed to bias the models that are comprised of shared features rather than idiosyncratic features to allow all datasets to be leveraged towards a single, well-powered model which ideally fits each individual domain. This increases the amount of data and trials input, thereby increasing the power of the model and its ability to classify trials.

Neuroimaging domain data is comprised of general subspace, specific subspace, and noise which is introduced by the scan and an individual's brain. Univariate fMRI analyses are able to derive the important aspect of the data as it averages activation across participants for the scan. Aggregating data over participants over time is not always the ideal route to test a

hypothesis. The use of a technique such as ridge regression can lend to the reduction of the excess signal as ridge shrinks the noise contributed by the scan, however, this analysis does not reveal the shared structure or determine the idiosyncratic signal. The $CoIR_{LS}$ algorithm shrinks the noise contributed by the scan and finds a set of weights that focuses on voxels that are shared across subjects, reducing the difference in domains and specific subspaces. This technique focuses on identifying the general subspace signal by minimizing the difference between domains.

The $CoIR_{LS}$ algorithm accounts for multiple sources of varied activation in participants' fMRI data while addressing the curse of dimensionality without aggregating across subjects. The algorithm is not perfect, however, and has a major limitation to be aware of in cognitive neuroimaging: the spatial link of voxels across subjects. Typically, fMRI data is analyzed across a subset of spatially relevant voxels (i.e., whole brain or an ROI) after subject scans are oriented to a standardized space, commonly MNI, which allows every subject's brain to be defined within the same boundaries from a set origin. While neural activity commonly occurs in concentrated brain space, subject brains are not a 1:1 link of voxel activation (D'Esposito et al., 2003; Handwerker et al., 2004; Kruggel & von Cramon, 1999), meaning that while fear may activate the insula in both subjects, insula voxel A of subject 001 may not activate identically to insula voxel A of subject 002. Brains being similar but not identical is why voxel activity is typically assessed across a subset of spatially relevant voxels. $CoIR_{LS}$ is unable to follow activation across spatially relevant voxels as in $CoIR_{LS}$, voxels are individually weighted based on their contribution to the classification and identifies voxels that correlate with a label across all participants, thereby regressing out some voxels during training. Results may be impacted by voxels being 1:1 linked across subjects. If a feature is unique to the target domain, it will be

difficult to identify it since the source domain considers one dimension while the target domain considers both dimensions. This limitation should be considered and mindful of in using the algorithm. The use of domain adaptation and $CoIR_{LS}$ remains a considerable step forward in fMRI analysis and the conclusions that can be drawn from the data, regardless of this limitation. Additionally, retaining the voxel-to-voxel correspondence allows the localization of the resulting function to be retained.

This dissertation found no significant differences in the ability of the $CoIR_{LS}$ algorithm to classify CS+ versus CS- fear acquired to viewed or imagined stimuli than ridge regression, a machine learning algorithm which uses the penalty lambda to reduce multicollinearity. The comparisons of the $CoIR_{LS}$ and ridge regression algorithms in the fear conditioning study were not found as anticipated, however, this does not mean the $CoIR_{LS}$ algorithm and domain adaptation should be abandoned, particularly as the algorithm has been shown to be beneficial and find shared representation across individuals (Zhou, 2022). The use of $CoIR_{LS}$ on this data has informed that the representation within fear conditioning in the whole brain is too different to be found in a shared subspace, meaning the representation either doesn't exist or excluded enough idiosyncratic signal that it is no longer able to be identified. The fear stimulus representation in a whole brain search is too different across individuals. The algorithm is beneficial depending on the data collected and the assumptions that are made. $CoIR_{LS}$ shares assumptions that are required with univariate analyses, such as the anatomical alignment across participants and the behavior of voxels. The assumptions made by $CoIR_{LS}$ are different from univariate analyses in the way univariate analyses aggregates signal over the space and voxels. The difference and similarities in assumptions being made when running the analyses indicate the way information is represented is addressed in each analysis differently. Significant

activation was found in the univariate analyses of the current study data, leading to the impression that effects should be able to be decoded, but not only do different standards of evidence give way to the signal detection in each analysis but additionally, cross-validation is used in CoIRLS that is not included in univariate analyses. Not only may the spatial smoothing of univariate analyses lend to finding the signal, but the cross-validation of $CoIR_{LS}$ may impact the signal finding as the statistical effect may not be prominent with any given holdout set. Future research will assess the use of $CoIR_{LS}$ in the current fear conditioning study using various ROIs, such as the amygdala, insula, and visual cortex.

This dissertation has highly lent to the progress of domain adaptation and $CoIR_{LS}$ use in cognitive fMRI research. The technical tutorial and GitHub of code make the CoIRLS algorithm accessible and easy to use for studies that predict a stimulus representation of a signal can be found in the shared subspace. Additionally, efforts of adapting the computer science-geared information and understanding of the $CoIR_{LS}$ and domain adaptation workings into the vocabulary of cognitive neuroscience will allow researchers to have a full understanding of how the algorithm works and the assumptions that are made in using the algorithm. This will help researchers know if the algorithm is fitting for their data and question at hand, rather than being uncertain or having a misunderstanding of the process and what can be obtained with the results.

Appendix A. CoIR Documents Created

$CoIR_{LS}$ Code

Utils.py

```
"""Collection of basic utility functions"""

import os
import itertools
import pandas as pd
import numpy as np
from typing import Iterable, List

def filter_matrix_by_set(row: pd.Series, var: str, by: str, set:
Iterable[str]):
    """ Apply a filter to matrices in each row of a data frame

    Parameters
    ----------
    row: pd.Series
    var: str
        The name of a variable in the data frame containing a 2-D
array (matrix)
        to be filtered row-wise.
    by: str
        The name of a variable in the data frame containing a 1-D
array or
        column vector of values that will be used to define the
filter.
    set: Iterable[str]
        A set of values that exist in `by`.

    Returns
    -------
    np.array

    """
    return row[var][[x in set for x in row[by]], :]
```

```
def load_npz_as_df(subjects: Iterable, roi: str, phase: str, exp:
str) -> pd.DataFrame:
    """Load npz files for a set of subjects as a data frame

    The npz files this function is intended to load are generated by
PyMPVA, and
    contain several objects:
    - samples (voxels)
    - sa.targets (+,-)
    - sa.target_extra (trial type),
    - sa.chunks (run)

    Parameters
    ----------
    subjects: Iterable
        A list of participant ids.
    roi: str
        An ROI label.
    phase: str
        A phase label.
    exp: str
        An experiment label.

    Returns
    -------
    pd.DataFrame
        All 1-D arrays are forced to be column vectors

    """
    d = pd.DataFrame(
        {
        "subject": subjects,
        "filepath": [
                os.path.join("data", "derivatives",
"00_roi_extraction", f"roi-{roi:s}", subj,
f"{subj:s}_phase-{phase:s}_exp-{exp:s}_roi-{roi:s}.npz")
                for subj in subjects
            ]
```

```
        }
    )

    NpzFiles = [np.load(f) for f in d.filepath]
    d["trial_types"] = [x["sa.target_extra"][:, np.newaxis] for x in
NpzFiles]
    d["stimulus_cond"] = [x["sa.targets"][:, np.newaxis] for x in
NpzFiles]
    d["runs"] = [x["sa.chunks"][:, np.newaxis] for x in NpzFiles]
    d["voxels"] = [x["samples"] for x in NpzFiles]
    [x.close() for x in NpzFiles]

    return d


def allzeros_across_all_runs(d: pd.DataFrame):
    runs = np.unique(d.runs_subset[0])
    z = []
    for i in runs:
        z.extend([np.all(y[x.flatten() == i, :] == 0, axis=0) for x,y
in zip(d.runs_subset, d.voxels_subset)])

    return np.any(np.array(z), axis=0).flatten()


def flatten_list(deep_list: List[List[object]]) -> List[object]:
    return list(itertools.chain.from_iterable(deep_list))
```

Modeling.py

```
import numpy as np
import pandas as pd
from tqdm import tqdm, trange
from collections import namedtuple
from typing import Callable, Union, List
from sklearn.metrics import accuracy_score, log_loss
from kale.pipeline.multi_domain_adapter import CoIRLS
```

```
DataCfg = namedtuple("DataCfg", ["target_field", "target_levels",
"data_field", "runs_field", "exclude_fold"])
Data = namedtuple("Data", ["X", "y", "C", "cv", "source",
"target_subject", "exclude_fold"])
HyperCfg = namedtuple("HyperCfg", ["alpha", "lambda_"])
Result = namedtuple("Result", ["target_subject", "cv_index",
"exclude_fold", "single", "acc_test", "acc_train", "loss_test",
"loss_train", "model_params", "model_weights"])


def cv_modelfit(fun: Callable[[Data, int], Result], d: pd.DataFrame,
single: bool, cfg: DataCfg, hyp: Union[HyperCfg,
List[List[HyperCfg]]]) -> pd.DataFrame:
    results = []
    for target_subject_index in trange(d.shape[0], desc="subject"):
        if single:
            data = pull_from_dataframe(d.iloc[[target_subject_index],
:], 0, cfg)
        else:
            data = pull_from_dataframe(d, target_subject_index, cfg)

        cv_set = np.unique(data.cv[~data.source])
        for cv_index in tqdm(cv_set, desc="cv", leave=False):
            if isinstance(hyp, HyperCfg):
                results.append(fun(data, cv_index, single, hyp))
            else:
                results.append(fun(data, cv_index, single,
hyp[target_subject_index][cv_index]))

    return pd.DataFrame(results)


def cv_coirls(d: pd.DataFrame, single: bool, cfg: DataCfg, hyp:
HyperCfg) -> pd.DataFrame:
    return cv_modelfit(run_coirls, d, single, cfg, hyp)


def cv_ridgels(d: pd.DataFrame, single: bool, cfg: DataCfg, hyp:
```

```
HyperCfg) -> pd.DataFrame:
    return cv_modelfit(run_ridgels, d, single, cfg, hyp)


def pull_from_dataframe(d: pd.DataFrame, target_subject_index: int,
cfg: DataCfg) -> Data:
    y_str = np.concatenate(d[cfg.target_field].values, axis=0)
    y = (y_str == cfg.target_levels[1]).astype(int)
    cv = np.concatenate(d[cfg.runs_field].values, axis=0).flatten()
    X = np.concatenate(d[cfg.data_field].values, axis=0)
    sub_index = np.concatenate(
        [[i]*x.shape[0] for i,x in enumerate(d[cfg.data_field])],
        axis=0
    )
    C = np.identity(d.shape[0])[sub_index,:]

    # Modify based on target subject (make sure target occupies
bottom rows)
    target_subject = d.subject.iloc[target_subject_index]
    min_cv = np.min(cv)
    z_source = sub_index != target_subject_index
    z_exclude = ((cv == cfg.exclude_fold) & ~z_source)

    X = X[~z_exclude, :]
    C = C[~z_exclude, :]
    y = y[~z_exclude]
    z_source = z_source[~z_exclude]
    cv = cv[~z_exclude]
    return Data(X, y, C, cv-min_cv, z_source, target_subject,
cfg.exclude_fold-min_cv)


def run_coirls(data: Data, cv_index: int, single: bool, hyp:
HyperCfg) -> Result:
    clf_ = CoIRLS(alpha=hyp.alpha, lambda_=hyp.lambda_)
    z_train = data.source | (data.cv != cv_index)
    X = (data.X - data.X[z_train, :].mean(axis=0)) / data.X[z_train,
:].std(axis=0)
    X_test = X[~z_train, :]
```

```
    X_train =X[z_train, :]
    y_test = data.y[~z_train]
    y_train = data.y[z_train]
    X = np.concatenate([X_train, X_test], axis=0)

    clf_.fit(X, y_train, data.C)
    y_train_pred = clf_.predict(X_train)
    y_test_pred = clf_.predict(X_test)
    return Result(
        data.target_subject,
        cv_index,
        data.exclude_fold,
        single,
        accuracy_score(y_test, y_test_pred),
        accuracy_score(y_train, y_train_pred),
        log_loss(y_test, y_test_pred),
        log_loss(y_train, y_train_pred),
        clf_.get_params(),
        clf_.coef_.numpy()
    )


def run_ridgels(data: Data, cv_index: int, single: bool, hyp:
HyperCfg) -> Result:
    clf_ = CoIRLS(alpha=hyp.alpha, lambda_=hyp.lambda_)
    z_train = data.source | (data.cv != cv_index)
    X = (data.X - data.X[z_train, :].mean(axis=0)) / data.X[z_train,
:].std(axis=0)
    X_test = X[~z_train, :]
    X_train =X[z_train, :]
    y_test = data.y[~z_train]
    y_train = data.y[z_train]
    C = np.ones([len(y_train), 1])

    clf_.fit(X_train, y_train, C)
    y_train_pred = clf_.predict(X_train)
    y_test_pred = clf_.predict(X_test)
    return Result(
        data.target_subject,
```

```
        cv_index,
        data.exclude_fold,
        single,
        accuracy_score(y_test, y_test_pred),
        accuracy_score(y_train, y_train_pred),
        log_loss(y_test, y_test_pred),
        log_loss(y_train, y_train_pred),
        clf_.get_params(),
        clf_.coef_.numpy()
    )
```

Datacheck.py

```
import numpy as np
from lb_dissertation.utils import load_npz_as_df
import pandas as pd
import os.path

phase = "B"
experiment = "task"
roi = "whole_bin"
target_levels = ("csp", "csm")
targets_label = "_".join(target_levels)
subj_df = pd.read_csv("participants.tsv", sep='\t')
subjects = subj_df["participant_id"]

d = load_npz_as_df(subjects, roi, phase, experiment)

for i in d['voxels']:
    for j in range(0,31):
        #print(d.loc[j]['subject'])
        if (len(d['voxels'][j]) != 72): print("trials")
        zerovals = np.all(d['voxels'][j], axis = 0)
        if (zerovals == True).all(): print(d.loc[j]['subject'] + "_"
+ str(j) + "_"  + "zero")
        for k in range(0, 72):
            fvals = len(d['voxels'][j][k])
            fn = np.isnan(d['voxels'][j][k])
```

```
            if (fn == True).any(): print(d.loc[j]['subject'] + "_"  +
str(j) + "_"  + "null")
            if (fvals != 193130): print(d.loc[j]['subject'] + "_"  +
str(j) + "_"  + "length voxs")
```

01_apply_roi.py

```
import numpy as np
from lb_dissertation.utils import load_npz_as_df
import pandas as pd
import os.path

phase = "B"
experiment = "task"
roi = "whole_bin"
target_levels = ("csp", "csm")
targets_label = "_".join(target_levels)
subj_df = pd.read_csv("participants.tsv", sep='\t')
subjects = subj_df["participant_id"]

d = load_npz_as_df(subjects, roi, phase, experiment)

for i in d['voxels']:
    for j in range(0,31):
        #print(d.loc[j]['subject'])
        if (len(d['voxels'][j]) != 72): print("trials")
        zerovals = np.all(d['voxels'][j], axis = 0)
        if (zerovals == True).all(): print(d.loc[j]['subject'] + "_"
+ str(j) + "_"  + "zero")
        for k in range(0, 72):
            fvals = len(d['voxels'][j][k])
            fn = np.isnan(d['voxels'][j][k])
            if (fn == True).any(): print(d.loc[j]['subject'] + "_"  +
str(j) + "_"  + "null")
            if (fvals != 193130): print(d.loc[j]['subject'] + "_"  +
str(j) + "_"  + "length voxs")
```

02_fit_models.py

```
import pandas as pd
import os.path
from itertools import product
from lb_dissertation.utils import load_npz_as_df,
filter_matrix_by_set, allzeros_across_all_runs
from lb_dissertation.modeling import cv_coirls, cv_ridgels, DataCfg,
HyperCfg
from itertools import product


phase = "B"
experiment = "task"
roi = "whole_bin"
target_levels = ("csp", "csm")
targets_label = "_".join(target_levels)
subj_df = pd.read_csv("participants.tsv", sep='\t')
subjects = subj_df["participant_id"]

alpha_set=[.001, .01, 1, 10, 100, 500, 1000]
# , .01, 1, 10, 100, 500, 1000
lambda_set=[.001, .01, 1, 10, 100, 500, 1000]

d = load_npz_as_df(subjects, roi, phase, experiment)

# Filter arrays within each row
for var in ["trial_types", "stimulus_cond", "runs", "voxels"]:
    d[f"{var:s}_tmp"] = d.apply(
        filter_matrix_by_set,
        var = var,
        by = "stimulus_cond",
        set = target_levels,
        axis = 1
    )

for var in ["trial_types_tmp", "stimulus_cond_tmp", "runs_tmp",
"voxels_tmp"]:
    lab = var.replace("_tmp", "")
```

```
    d[f"{lab:s}_subset"] = d.apply(
        filter_matrix_by_set,
        var = var,
        by = "trial_types_tmp",
        set = ("image", "view"),
        axis = 1
    )

z = allzeros_across_all_runs(d)
d.voxels_subset = [x[:,~z] for x in d.voxels_subset]

d = d.drop(["trial_types_tmp", "stimulus_cond_tmp", "runs_tmp",
"voxels_tmp"], axis = 1)

cfg = DataCfg(target_field="stimulus_cond_subset",
target_levels=target_levels, data_field="voxels_subset",
runs_field="runs_subset")
HyperCfgs =[HyperCfg(alpha=x, lambda_=y) for x,y in
product(alpha_set, lambda_set)]
r = []
for hyp in HyperCfgs:
    r.append(cv_coirls(d, False, cfg, hyp))
    r[-1].loc[:, "model_type"] = "coirls"
    r[-1].loc[:, "cfg"] = [cfg]*r[-1].shape[0]
    r[-1].loc[:, "hyp"] = [hyp]*r[-1].shape[0]


HyperCfgs =[HyperCfg(alpha=x, lambda_=y) for x,y in
product(alpha_set, [0])]
for hyp in HyperCfgs:
    r.append(cv_ridgels(d, True, cfg, hyp))
    r[-1].loc[:, "model_type"] = "ridgels"
    r[-1].loc[:, "cfg"] = [cfg]*r[-1].shape[0]
    r[-1].loc[:, "hyp"] = [hyp]*r[-1].shape[0]

    r.append(cv_ridgels(d, False, cfg, hyp))
    r[-1].loc[:, "model_type"] = "ridgels"
    r[-1].loc[:, "cfg"] = [cfg]*r[-1].shape[0]
    r[-1].loc[:, "hyp"] = [hyp]*r[-1].shape[0]
```

```
R = pd.concat(r)
R.drop(["model_params", "model_weights", "cfg"], axis=1).to_csv(
    os.path.join("results",
f"phase-{phase:s}_exp-{experiment:s}_roi-{roi:s}_dv-{targets_label:s}
_hyperconfigs-coir.csv")
)
R.to_pickle(
    os.path.join("results",
f"phase-{phase:s}_exp-{experiment:s}_roi-{roi:s}_dv-{targets_label:s}
_hyperconfigs-coir.pkl")
)
print(R)
```

Appendix B. Domain Adaptation and CoIR Guide

Introduction

Contents

Cognitive Neuroscience

The goal of cognitive neuroscience is to relate cognitive processes and human behavior to neural processes. While this initially took the form of mapping cognitive processes to localized brain structures, this has given way to a network-distributed perspective: cognitive processes are supported by combinations of potentially anatomically distinct brain regions. For example, working memory relies on frontal, parietal, and modality specific regions working in tandem, and self-referential thinking taps the default mode network. Even within a single region, the relationship between brain activity and behavior can be complex. Stimuli from a wide range of stimulus domains can be decoded from the same patch of cortex in the ventral temporal lobe, for example. What differentiates the neurocognitive representations of "shoe" and "chair", for example, is not where the brain is active, but the specific pattern of activation. Furthermore, the neurocognitive representation of a stimulus, concept, or process may not be anatomically contiguous, and some neural units that contribute to the representation may contribute more than others. Identifying relationships between the brain and behavior is thus a serious challenge.

Using Machine Learning

In attempts to overcome this challenge, many cognitive neuroscientists have begun to experiment with techniques developed for machine learning, including regularized regression, support vector machines, and deep neural networks, to learn models that aim to decode patterns of neural activity into the condition or stimulus labels that they are associated with. Machine learning is not magic, however. To learn an accurate model requires a large number of labeled stimuli or conditions; ideally, there would be more labeled examples than there are neural features, but this is very rarely the case in cognitive neuroscience where it is typical to model each subject individually.

Ideally, this could be combated by combining the data from multiple subjects in the same data. The simplest strategy involves anatomically aligning all subjects to the same template and concatenating all the functional data by trial. This is similar to what is done in some univariate analyses of fMRI data, but it's not ideal for multivariate decoding studies. To do so would assume that voxels at the same standard-space coordinate across subjects share a common response profile, and this is known to not be the case. Before aggregating across subjects–or even potentially across studies–the datasets should be functionally aligned, such that all datasets are described with respect to the same functional features.

Domain Adaptation

Again, we can borrow from the machine learning literature to attempt to achieve this goal. In what follows, I will introduce a computational technique called domain adaptation which has already proved promising for aligning fMRI datasets from multiple subjects, collected across multiple sites and scanners, and even across studies, to effectively increase the training set used for machine learning. When target and source domains share key elements of functional variance relevant to the machine learning classification problem, regardless of how that

functional variance is expressed in the voxel-space of each subject, model fits can improve. I will begin with an overview of machine learning before explaining what domain adaptation is, how it can be used in cognitive neuroscience, and when and why it can be applied.

By Lauryn Burleigh

Machine Learning

Contents

Machine learning (ML), a category of artificial intelligence, is a useful tool for multiple domains but can be particularly impactful for cognitive neuroscientists and functional magnetic resonance imagining (fMRI) researchers as it enables computers to learn a task based on data given, similar to how humans learn from experience, without explicitly programming the learning task. The process allows computers to modify their actions over multiple iterations, or attempts, to improve accuracy (i.e., number of times the action produces a correct result). The machine learning process is related to Bayesian statistics, a null hypothesis statistic testing alternative which allows prior knowledge to influence the test at hand, and computational statistics, which make statistical predictions via computers. Solving a problem via traditional ML involves a model (i.e., the task to learn or problem to solve) being trained on a dataset to learn the task question at hand, and being tested on alternate but identical data (i.e., a subset of the original dataset: data of one run from each subject, or one subject, of the study used in training but not included in the training itself) (Alzubi et al. 2018). An important requirement of traditional ML is that a dataset must include the same data for the learning task (i.e., data from only one study can be used in the model).

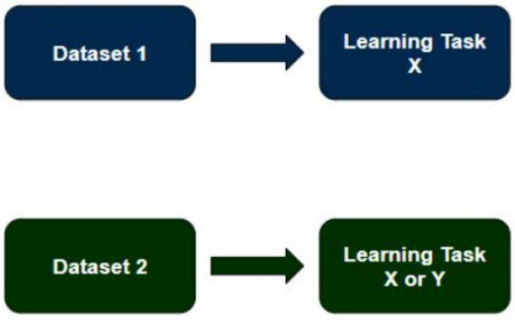


Fig. 1 - Traditional ML allows a dataset from a single study to contribute to the model's learning task.

Questions of Machine Learning

Prior to solving the problem, the appropriate ML algorithm must be chosen by determining which one (or more) of the five questions data science can answer that the problem at hand fits into: classification problem (e.g., Is this A or B?), anomaly detection problem (e.g., Which is the odd one out?), regression problem (e.g., How much/many?), clustering problem (e.g., What is the structure/organization?), or a reinforcement learning problem (e.g., What should be done next?).

Classification

The majority of cognitive neuroscience ML algorithms are classification problems in which a fixed number of output classes are determined a priori, such as Yes/No, but are not required to be binary (Alzubi et al. 2018). Some common classification problems in cognitive neuroscience are fear conditioning conditioned stimulus classification (i.e., Is the trial a feared conditioned stimulus?), percept modality classification (i.e., Is the trial viewed or imagined/viewed or heard/etc.?), and category classification (i.e., Which animal/tool/etc. is present in the trial?). Multi-voxel Pattern Analysis (MVPA) is a common fMRI traditional machine learning classification tool.

Anomaly Detection

An anomaly detection problem detects changes or anomalies (i.e., outliers) in a pattern (Alzubi et al. 2018). For example, in cognitive neuroscience it has been used to discriminate seafarers with altered default mode network functioning from healthy controls (Shi et al. 2015).

Regression

Regression problem algorithms handle continuous, numeric output, generally asking "How much?" or "How many?" (Alzubi et al. 2018). Cognitive fMRI researchers have used regression ML analyses for many purposes, from predicting ratings of features experienced in video games (Di Bono and Zorzi 2008) to predicting alcohol misuse based on neuropsychological profiles (Whelan et al. 2014) to predicting math abilities and associated neural mechanisms following tutoring (Supekar et al. 2013).

Clustering

A clustering problem is one which creates clusters based on the similarity of the data structure. Due to the large number of features in fMRI data (i.e., many voxels), clustering algorithms are very commonly used to reduce the number of important features, such as independent components analysis (ICA) and principle component analysis (PCA) (Smolders et al. 2007; Liang et al. 2016), and are often used as a precursor in further machine learning algorithms (Alzubi et al. 2018).

Reinforcement

Finally, reinforcement problem algorithms learn the behavior/task to solve through trial and error in a continuously changing environment, using defined rewards and penalties (Alzubi et al. 2018). Though reinforcement algorithm models have not been widely used in cognitive neuroscience, one has been used to predict neural activation in various cortico-striatal loops during stimulus-action-reward association learning (Haruno and Kawato 2006).

Transfer learning (TL) can be used for all five questions, but this guide focuses on classification as it is the most common for cognitive neuroscience fMRI ML models.

By Lauryn Burleigh

Transfer Learning and Domain Adaptation

Contents

Transfer Learning (TL) addresses the limitation of knowledge generalization and sharing in traditional ML models. Rather than allowing only identical data (i.e., one study) to contribute to a model, TL can leverage information gained from related datasets, allowing a larger and more diverse set of data to influence the model. This allows TL to more similarly represent humans than traditional ML, as humans generalize knowledge learned to new situations both intentionally (Pfordresher et al. 2021) and unintentionally (Greening et al. 2021).

Source and Target Domains

Data input into TL is either part of the source domain (i.e., the data to be leveraged; blue dataset box in Fig. 3a) or target domain (i.e., the data to be classified; green 'Dataset' box in Fig. 3a). The task to be learned, which uses the source domain data, is the source task (blue 'Learning Task' box in Fig. 3a). Using the source domain data to train the source task results in the model gaining the knowledge of this learned source task. This knowledge-improved model then influences the target task (white 'Learning Task' box in Fig. 3a) which the target domain data is used for. When the same source and target domains are used (i.e., identical data from one study), and the same source and target tasks are used (i.e., one learning task), the analysis is equivalent to a traditional ML model.

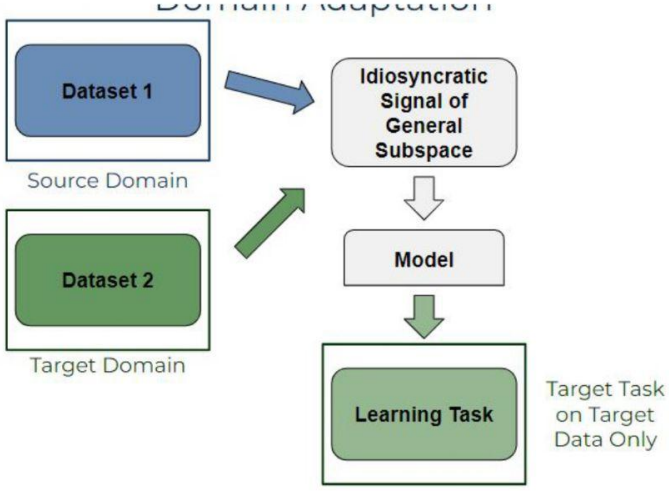


Fig. 2 - Domain Adaptation process.

Types of Transfer Learning

There are three subtypes of TL based on the relationship between the two domains and between the two tasks (see Table 1):

- Inductive TL uses the same domains and different but related tasks
- Unsupervised TL uses different but related domains and different but related tasks
- Transductive TL uses different but related domains and the same tasks

A domain has two components: a feature space (i.e., the space of all term vectors) and a marginal probability distribution of the learning sample, both of which are derived from the same dataset. When the source and target domain are different, it implies that either the source and target feature spaces, or the source and target probability distributions are not the same.

Domain adaptation (DA), the focus of this guide, specifies that the difference in domains occurs in the marginal probability distributions. Transductive TL, and therefore domain adaptation, specifies that *the learning tasks are the same*, implying that a predictive function learned in the source domain can be adapted to use in the target domain if some unlabeled target-domain data is provided.

Learning Type		Source + Target Domains	Source + Target Tasks
Traditional Machine Learning		Same	Same
Transfer Learning	Transductive Transfer Learning [i.e., Domain Adaptation]	Different (but related)	Same
	Inductive Transfer Learning	Same	Different (but related)
	Unsupervised Transfer Learning	Different (but related)	Different (but related)

Table 1 - The relationship between traditional ML and the three TL subtypes.

By Lauryn Burleigh

Domain Adaptation

Contents

When to use Domain Adaptation

Before determining the types of TL approaches that can and should be used for domain adaptation specifically, which situations transferring skills should be used and should not be used needs to be assessed.

DA should not be used if the source and target domains are not related. For a source and target domain to be related, an explicit or implicit relationship must exist between the feature spaces of the two domains (e.g., same study methodology at different locations, same study methods only altering stimuli, similar questions addressed in studies, etc.).

A brute-force transfer with domains that are not related to each other may be unsuccessful, and at worse, may hurt the learning performance in the target domain resulting in negative transfer.

Domain Adaptation Limitation

The use of domain adaptation has an important limitation to be aware of. Unlike other fMRI analyses, domain adaptation must retain a spatial link of voxels across subjects.

Typically, fMRI data is analyzed across a subset of spatially relevant voxels (i.e., whole brain or a region of interest) after subject scans are similarly oriented to a standardized space, commonly MNI as used in the study below, which allows every subject's brain to be defined within the same boundaries from a set origin. While activity may commonly occur in concentrated brain space, subject brains are not a 1:1 link of voxel activation, meaning that while fear may activate the insula of both subjects, insula voxel A of subject 001 may not activate identically to insula voxel A of subject 002. Voxel activity is typically assessed across a subset of spatially relevant voxels because brains are similar but not identical.

Domain adaptation is unable to follow this rationale as voxels are individually weighted based on their contribution to the classification and identifies voxels that correlate with a label across all participants and therefore some voxels will be regressed out during training. The need for voxels to be 1:1 linked across subjects may impact the results as if a feature is unique to the target domain, it will be difficult to identify it as the source domain considers one dimension while the target domain considers both dimensions.

While this limitation should be thoughtfully considered, domain adaptation remains a considerable step forward in fMRI analysis and the conclusions that can be drawn from the data. Additionally, if voxel to voxel correspondence is given up in domain adaptation, localization of function is also abandoned as the voxel space is.

CoIR

Traditional domain adaptation considers each dataset to be a different domain. This is beneficial and widely used in computer vision and natural language processing, however brain activity presents a particular challenge as participants can elicit different patterns of activity for the same cognitive process.

In order to account for this, the current guide uses the Covariate Independence Regularization (CoIR) framework in which each subject is considered a unique learning task and extracts subject-specific features. Treating each dataset as a different domain as well as each subject as a different domain allows for a more robust model with each unique experiment-subject combination as a unique domain.

By Lauryn Burleigh

About CoIR

The CoIR algorithm considers a target dataset with labeled and unlabeled samples acquired from one or more source datasets, where the objective learning task is to predict the brain conditions of unlabeled target samples (cite diss).

Both the target and source experiments must have the same brain conditions to classify. To classify the conditions, the framework optimizes three objectives:

- minimizing empirical error on labeled data
- minimizing dependence on domain covariates
- minimizing model complexity (i.e., reduce the upper bound of generalization risk).

The classifiers can be viewed as feature mapping where samples of all domains are in the same distribution and project the input features to a one-dimensional output space, regularizing the hypothesis space to reduce model complexity. Additionally, the algorithm minimizes empirical error on labeled data and domain dependence on domain covariates further reducing model complexity. Minimalizing the prediction/empirical error and the domain dependence occurs in the framework simultaneously, so can therefore be viewed as combining the benefits of domain-invariant classifier methods and domain dependence minimization mapping.

The CoIR_{LS} algorithm uses the regularized least squares classifier, considering unique experiment-subject combinations by encoding a matrix of experiment designs and subjects as domain covariates for domain adaptation, called one-hot encoding. The output is a one-dimensional space of model weights for the classification task at hand, which can be used to identify the top 1% coefficients (in magnitude) to in turn visualize the brain areas of positive and negative activation for interpretation.

By Lauryn Burleigh

$CoIR_{LS}$

Contents

The $CoIR_{LS}$ framework is completed in 4 steps:

1. encode domain covariates with one-hot encoding
2. construct the identity matrix, centering matrix, probability matrix, and a matrix denoting unlabeled target data as 0s
3. construct kernel matrix and linear kernel
4. compute and return the classifier based feature mapping.

Input

Inputs required include:

- data matrix of source and target data
- vector of training labels
- domain covariates

Hyper-Parameters

Hyper-parameters (i.e., values that control the model's learning process) include:

- lambda (λ), which controls the importance of dependence on domain covariates
- alpha (α), which controls the constraint of coefficients

Optimal hyper-parameters can be found using leave-one-domain-out cross validation on source domain data.

By Lauryn Burleigh

CoIR Code

Contents

All scripts and code to run the $CoIR_{LS}$ algorithm can be access on GitHub CoIR

Directory Setup

Store functional nifti fMRI files in the following directory: data > raw > pymvpa > **subject#** > **fmri_files**.nii.gz

ROI binary group masks should be stored in the directory: data > raw > roi > **mask_name**.nii.gz

Script 1

Script 01_apply_roi.py should be run first. This script takes in the functional fMRI nifti files and a binary ROI mask in order to create a final .npz which stores the relevant task, mask, and fMRI information to be used in the following algorithm.

First, edit the variables assigned in lines 13-18 of the 01_apply_roi.py script.

- Line 13 denotes that the condition tuple contains a phase, experiment, mask, and subject.
- Line 16 defines the phases. In the study this code was designed for, the two phases are two sets of tasks that were performed by each participant.
- Line 17 defines the experiments. In the study the code was designed for, the experiments are the two task types given in each phase. The "hab" experiment refers to a habituation task in which the instructions were learned and performed while the "task" experiment refers to the main task in which fear conditioning was introduced.
- Line 18 defines the mask name **wihtout** .nii.gz
- Lines 13 and 19-23 may need to be edited if any of the previous variables are not relevant to your study.

When all variables are adjusted to the study data, run this python script.

Script 2

Script 02_fit_models.py takes in the .npz file created from 02_fit_models.py to run the $CoIR_{LS}$.

First, edit the variables assigned in lines 7-19 of the 02_fit_models.py script.

- Line 7 specifies the phase to run the algorithm on
- Line 8 defines the task to run the algorithm on
- Line 9 defines the ROI to run the algorithm on, again leaving out .nii.gz of the filename
- Line 10 defines the classification trials of interest. In the study this code was created for, the goal was to classify whether each trial was a "csp" or "csm"
- Line 15 defines the alpha hyperparameter. Multiple alpha values can be provided in a list for this variable to perform a hyperparameter search, identifying the ideal value

CoIR Output

The resulting file from the 02_fit_models.py script is a single .csv, named with the variables defined in the script for the study.

The result CSV will contain an accuracy and loss value obtained by the training and testing. The 4 values will be obtained for each algorithm run. Each run of each subject for each alpha-lambda hyper parameter combination of each CoIR, ridge with all subjects, and ridge with a single subject, will be output.

This output CSV can be analyzed in the statistical software of your choice, as any other data file.

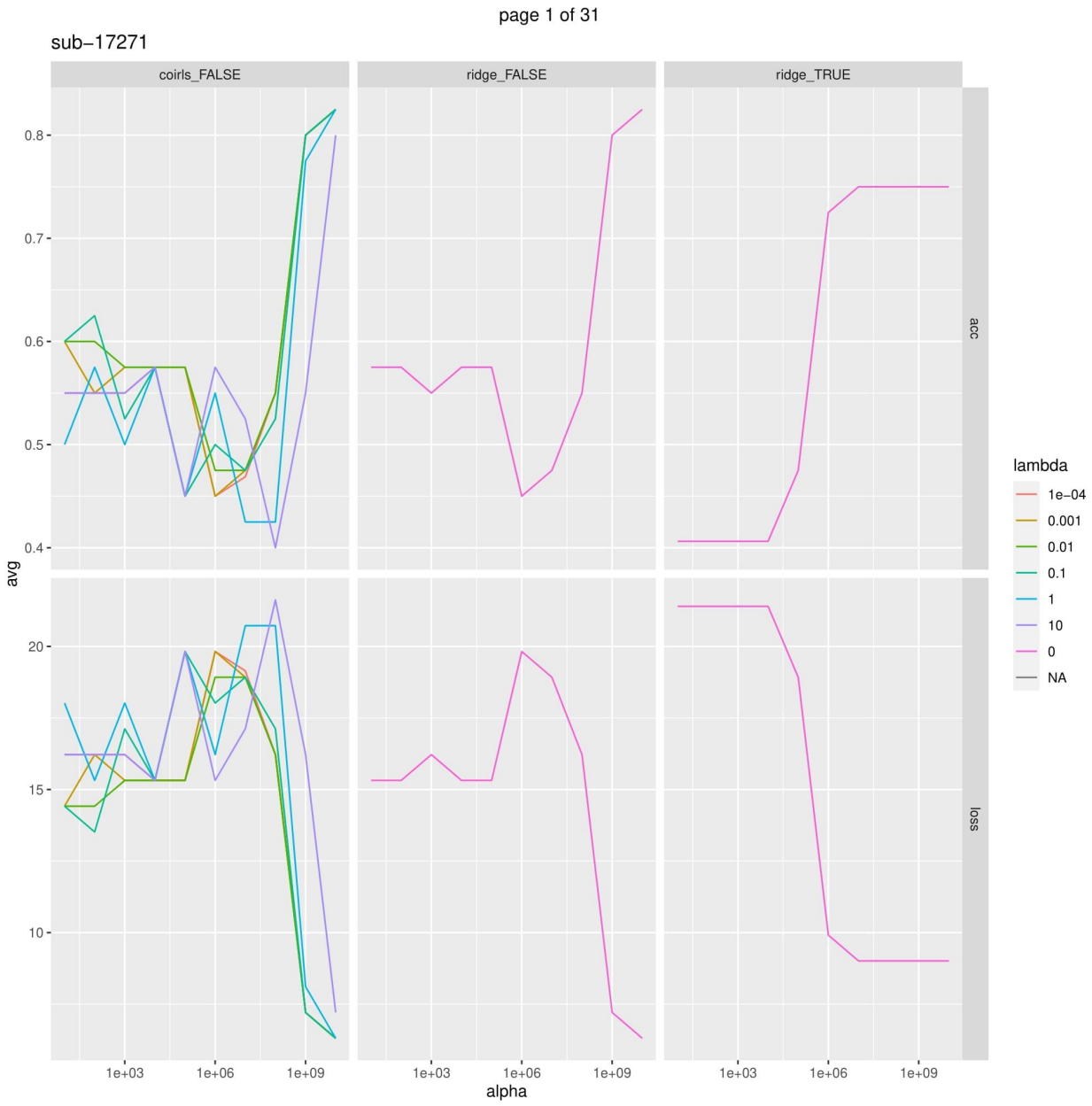
By Lauryn Burleigh

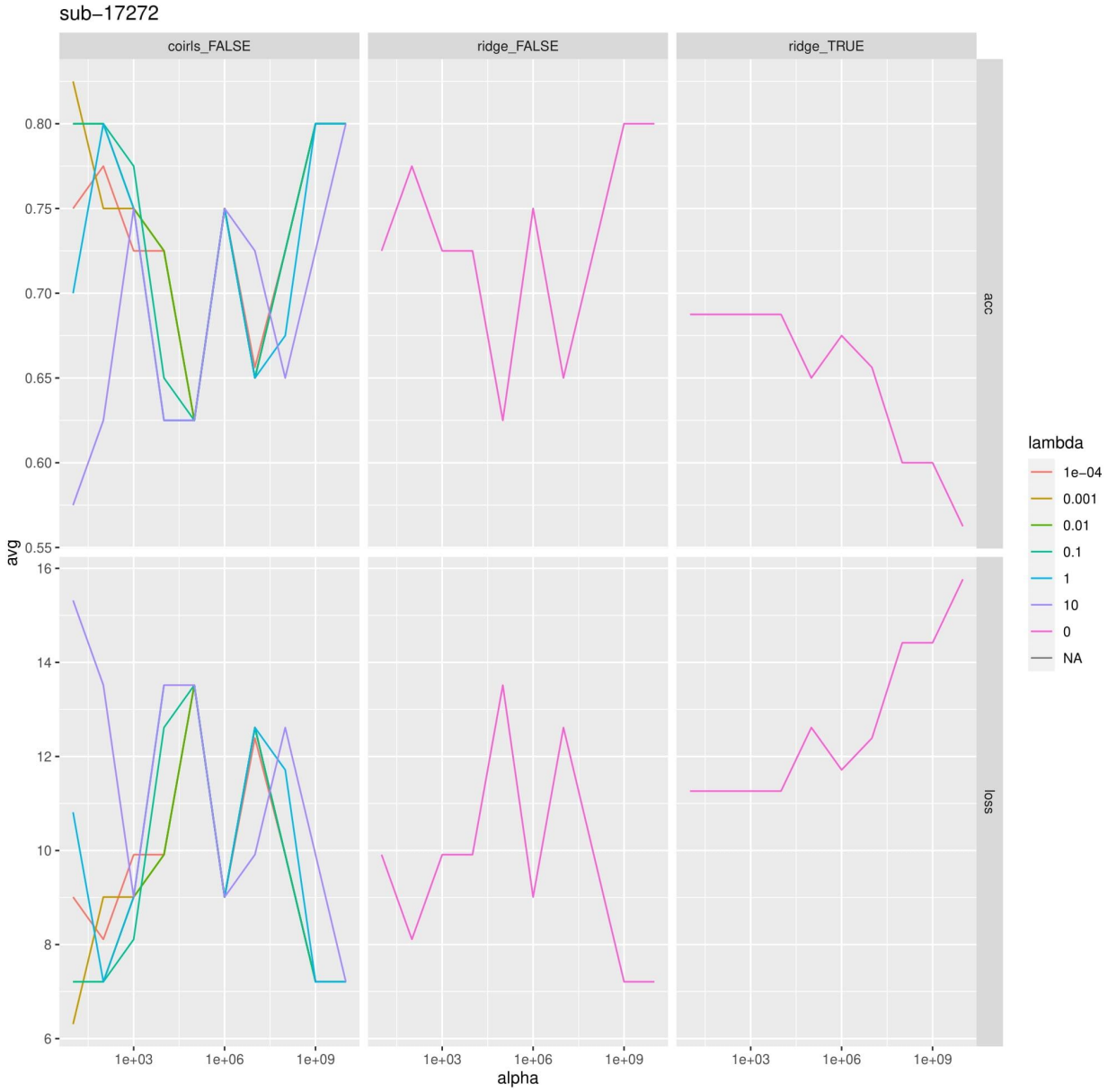
- Line 17 defines the lambda hyperparameter. Multiple lambda values can be provided in a list for this variable to perform a hyperparameter search, identifying the ideal value
- Line 19 defines the variables that are included in the filename of the .npz. If no variables were removed from the previous script (01_apply_roi.py), no variables will need to be removed or changed from this line.

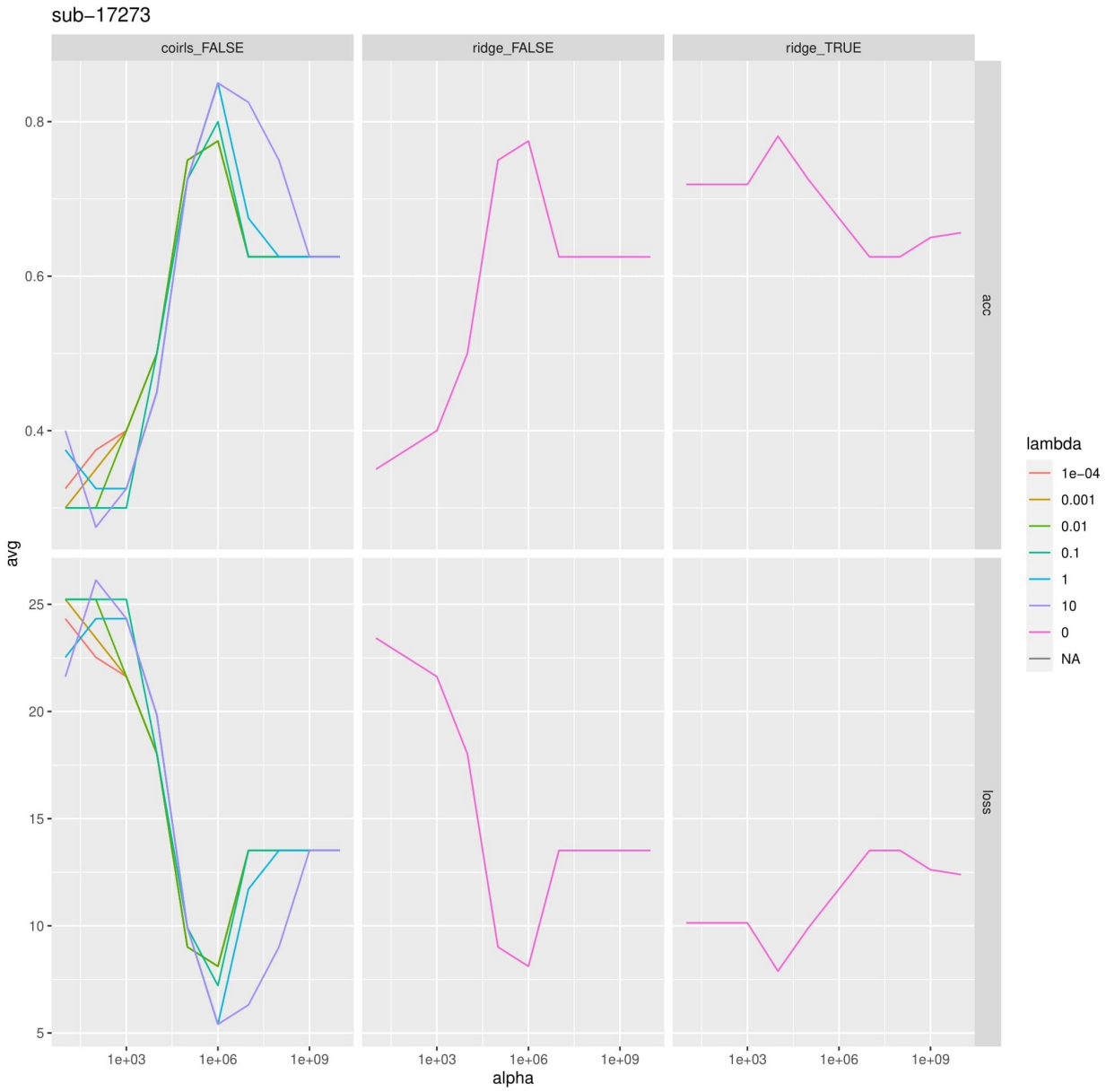
When all variables are defined appropriately for the study data at hand, run the 02_fit_models.py script.

By Lauryn Burleigh

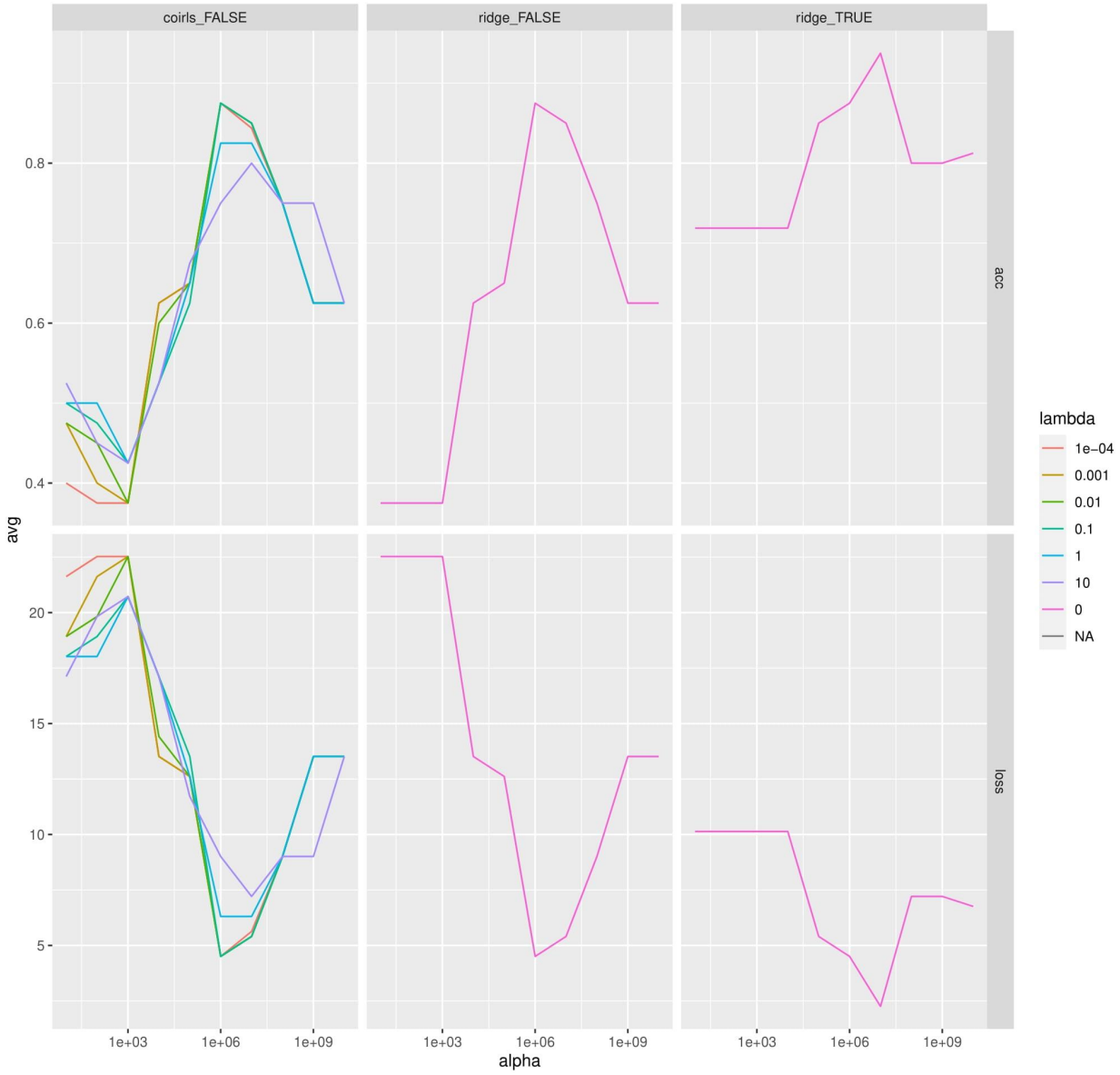
Appendix C. Hyperparameter Search for Each Subject

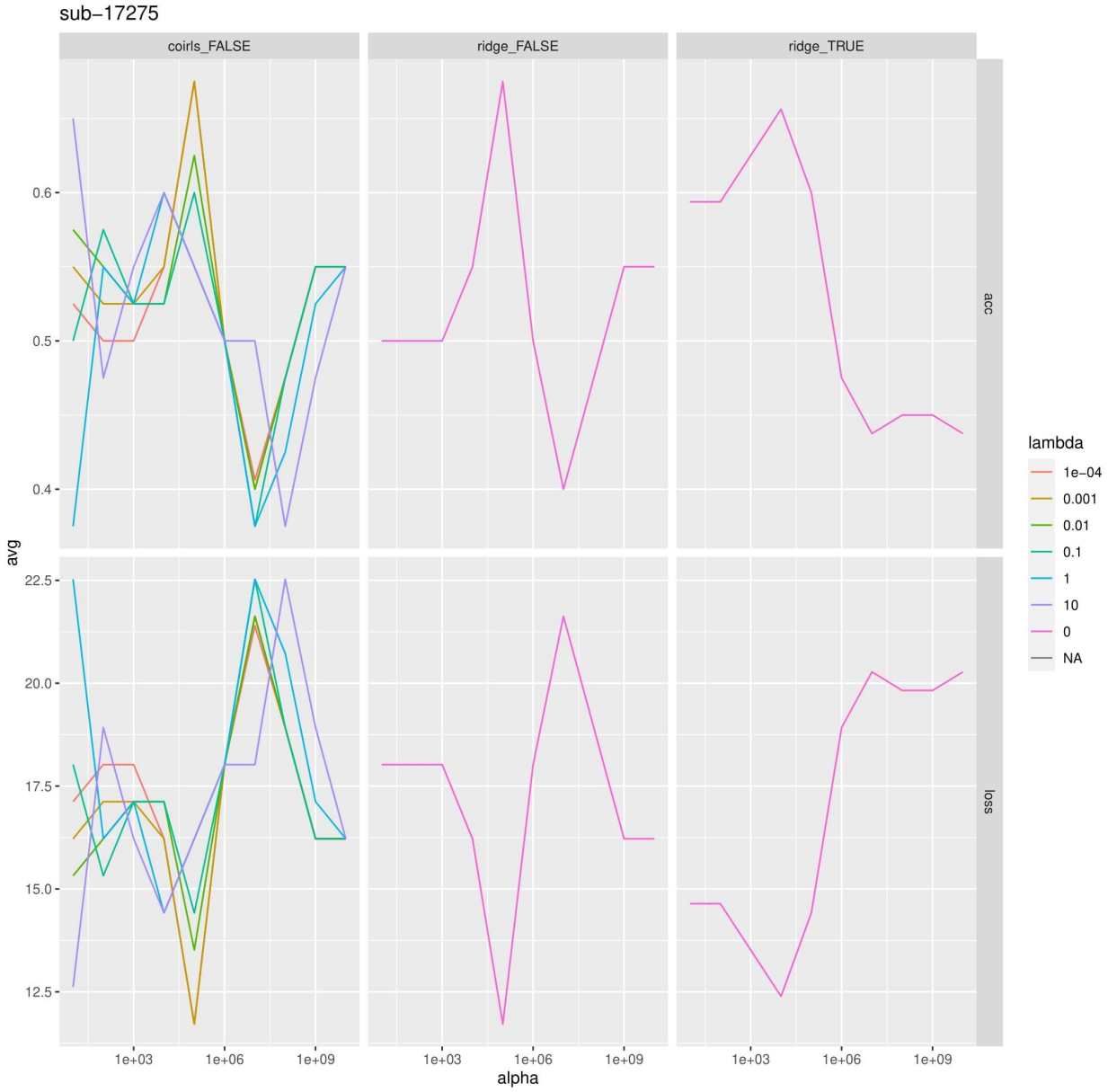


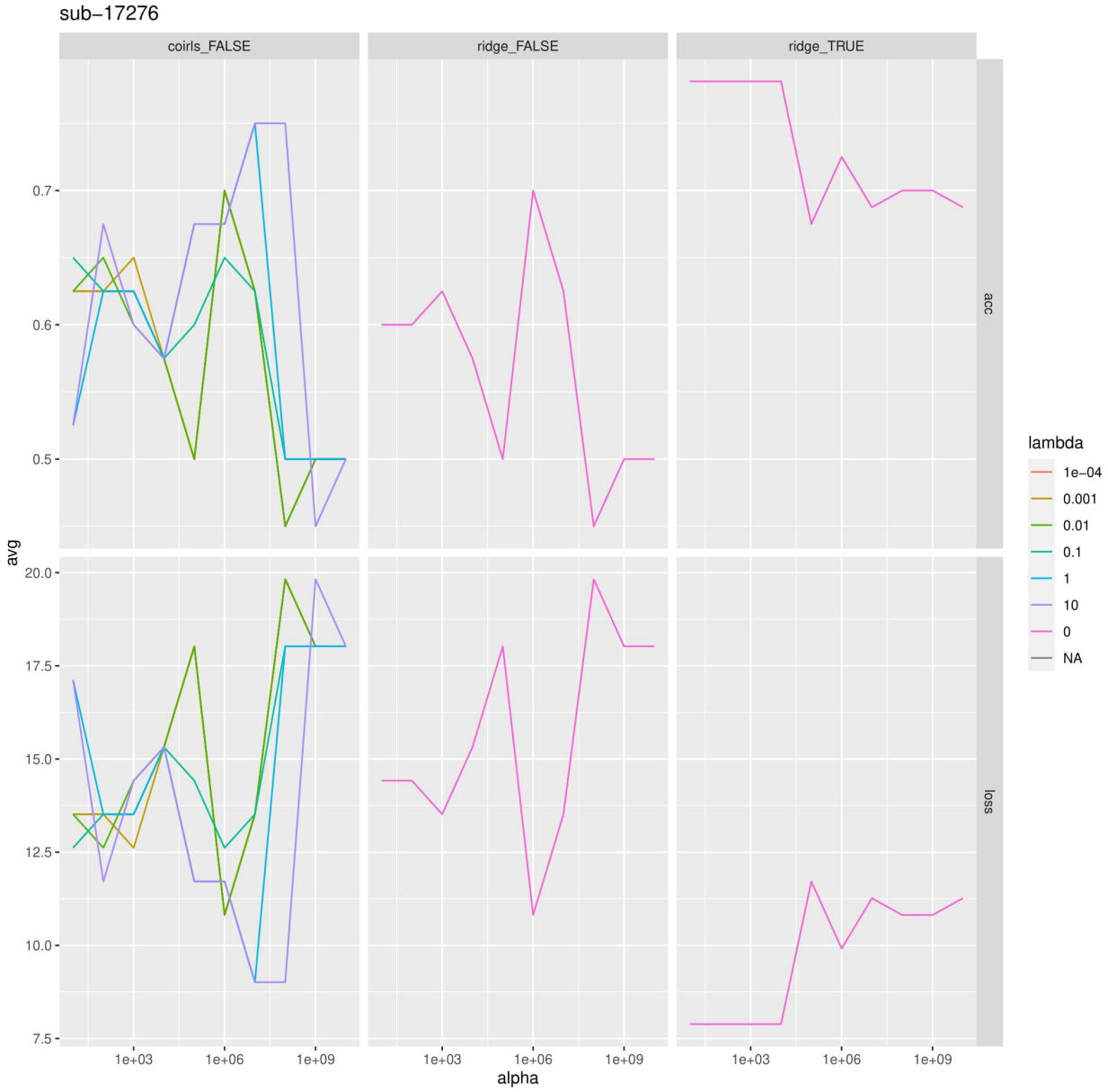

sub−17272
coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.80
0.75
0.70
0.65
0.60
0.55
16
14
12
10
8
6
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

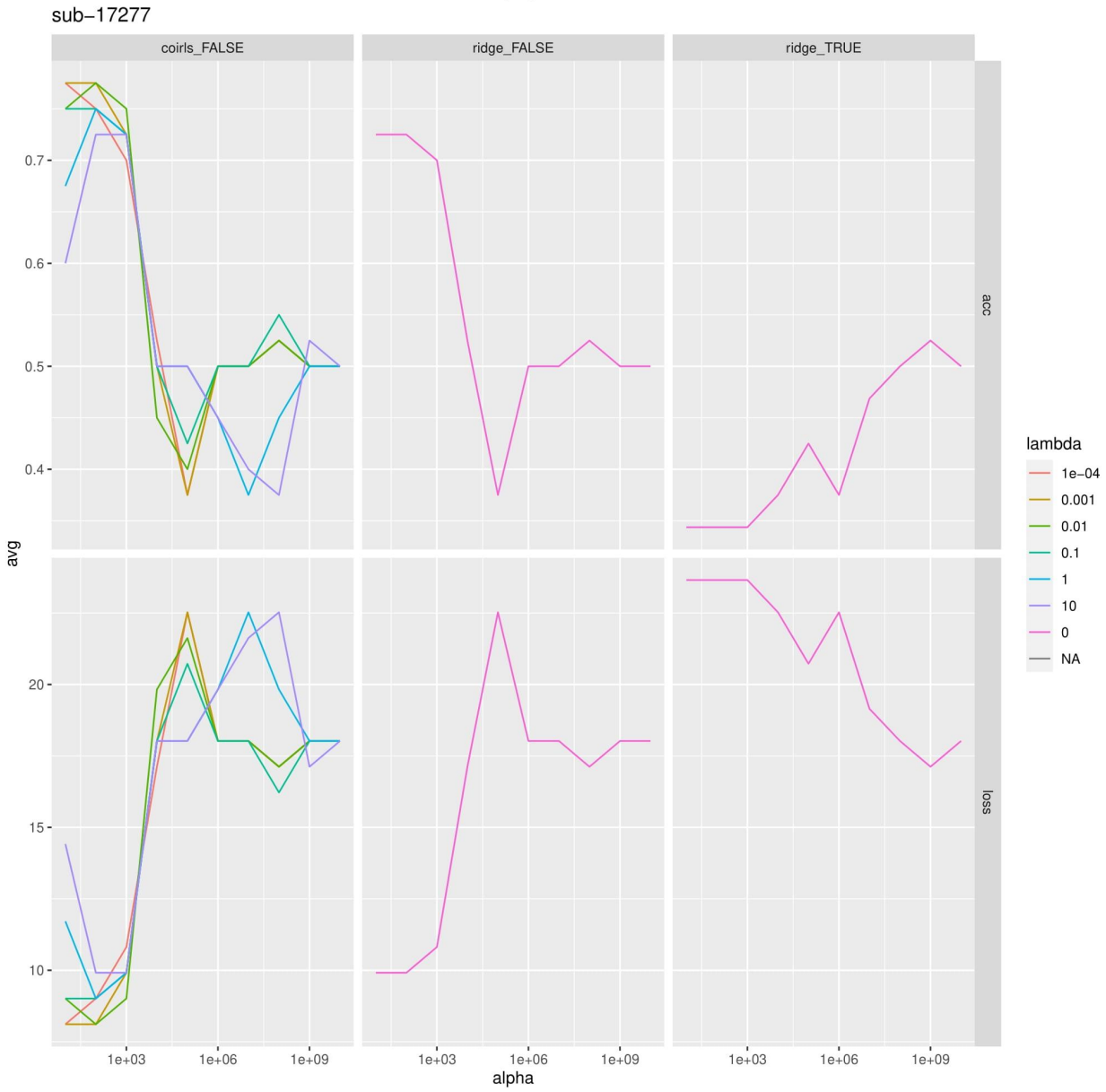

sub−17273
coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.8
0.6
0.4
25
20
15
10
5
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

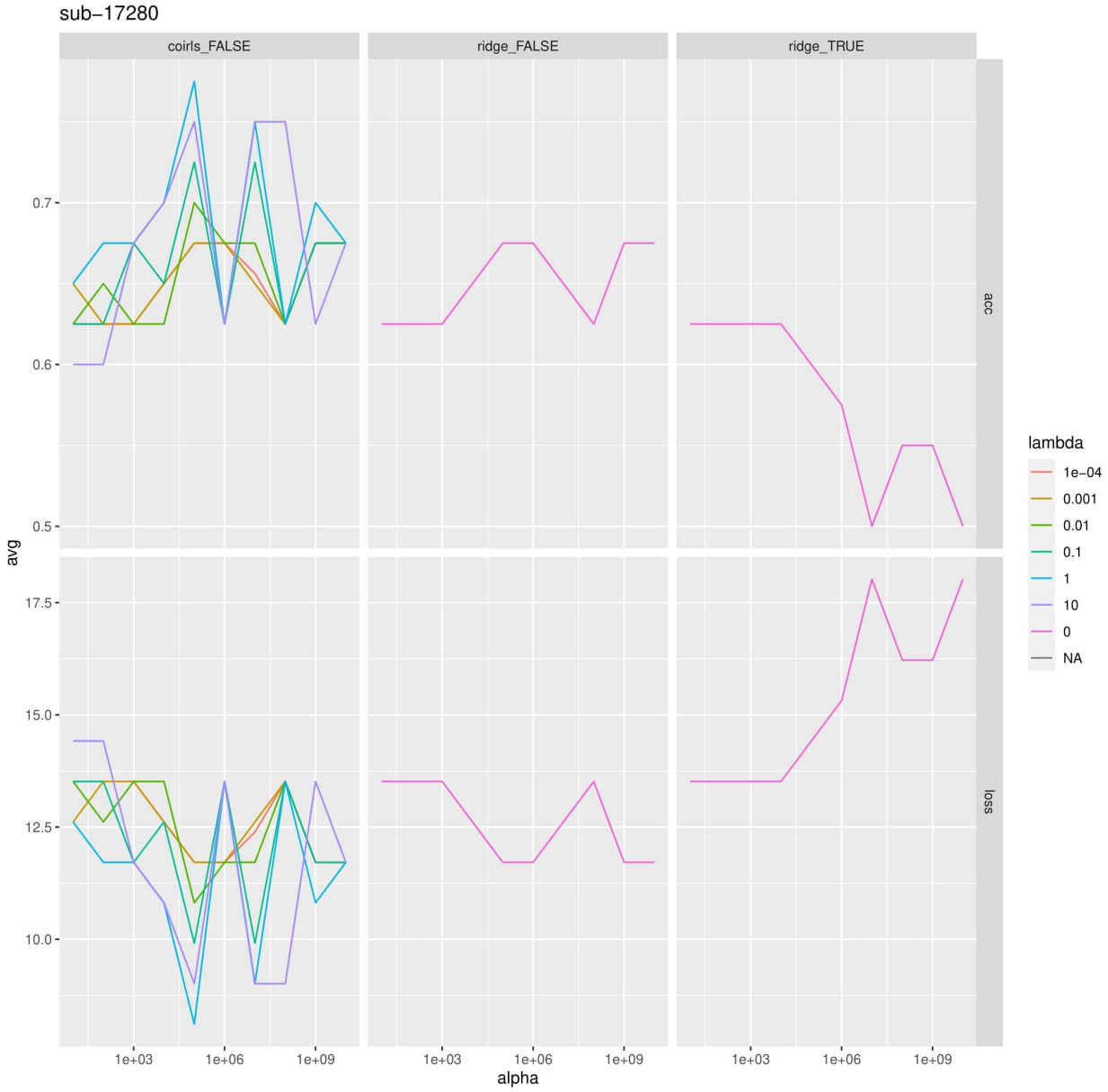
sub−17274


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
0.8
0.6
0.4
20
15
10
5
1e+03
1e+06
1e+09
alpha
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

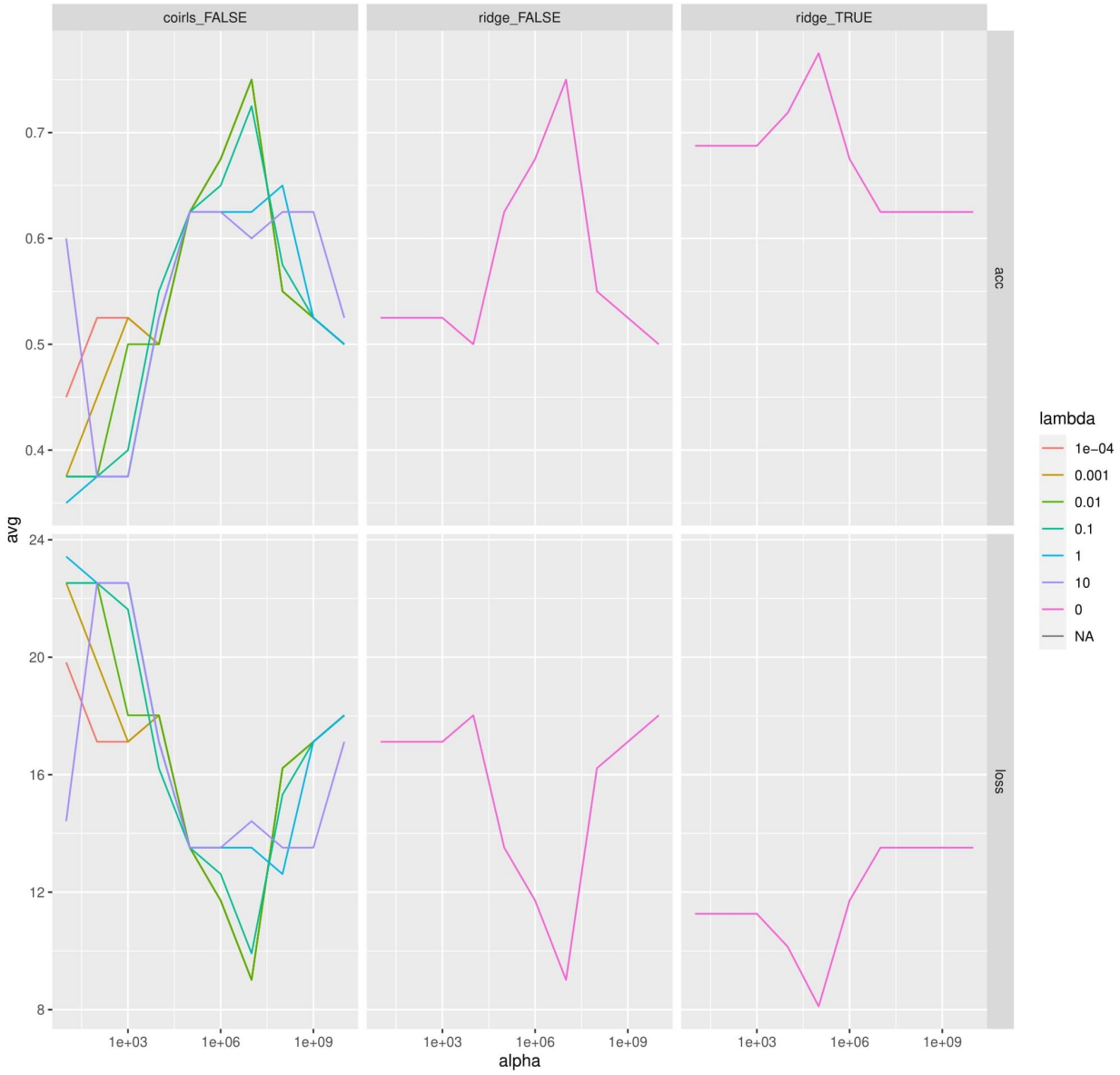

sub−17275
coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.6
0.5
0.4
22.5
20.0
17.5
15.0
12.5
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA


sub−17276
coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.7
0.6
0.5
20.0
17.5
15.0
12.5
10.0
7.5
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

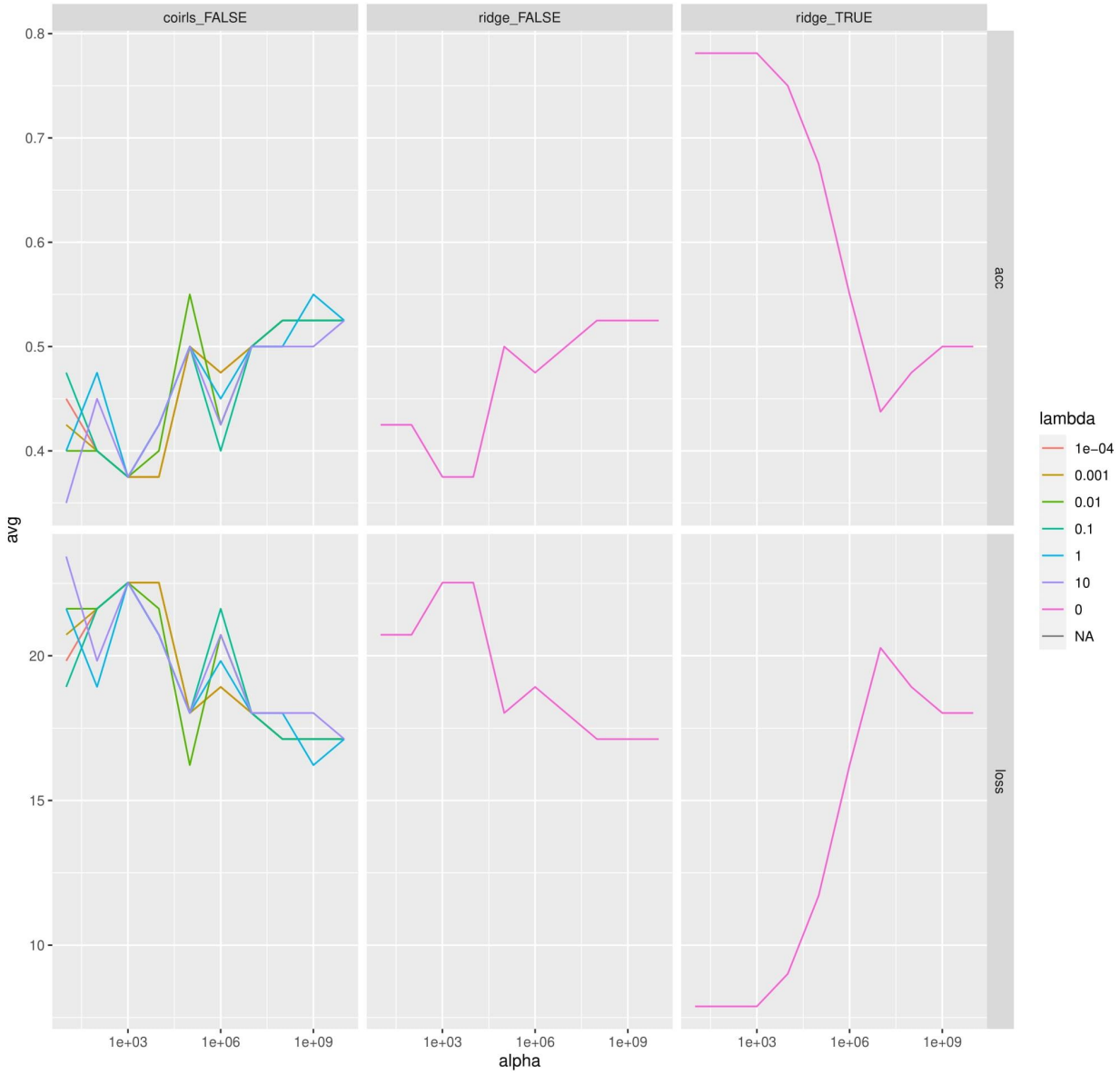

sub−17277
coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.7
0.6
0.5
0.4
20
15
10
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA


sub−17280
coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
0.7
0.6
0.5
17.5
15.0
12.5
10.0
1e+03
1e+06
1e+09
alpha
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

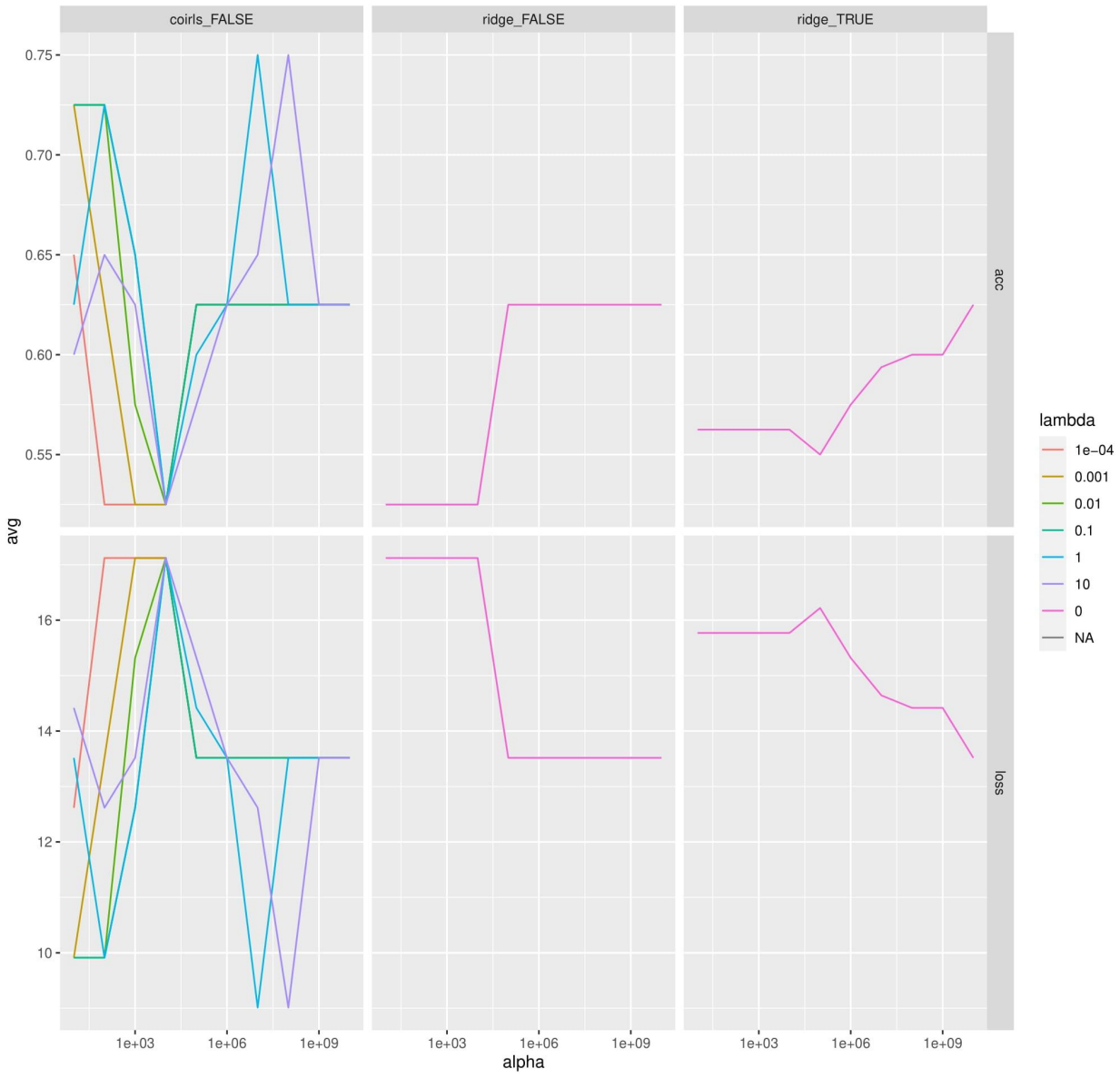
sub−17288


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.7
0.6
0.5
0.4
24
20
16
12
8
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

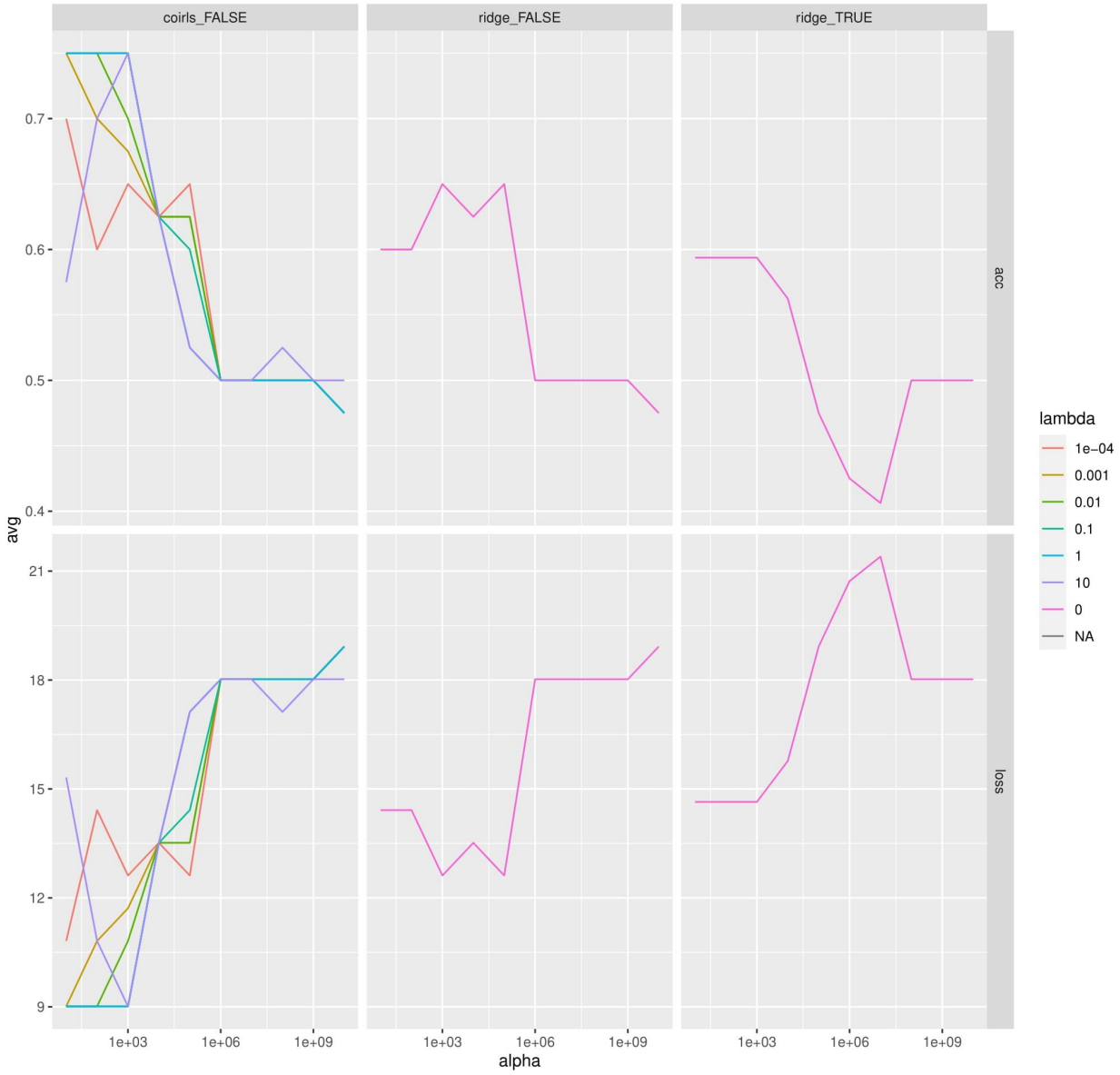
sub−17299


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.8
0.7
0.6
0.5
0.4
20
15
10
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

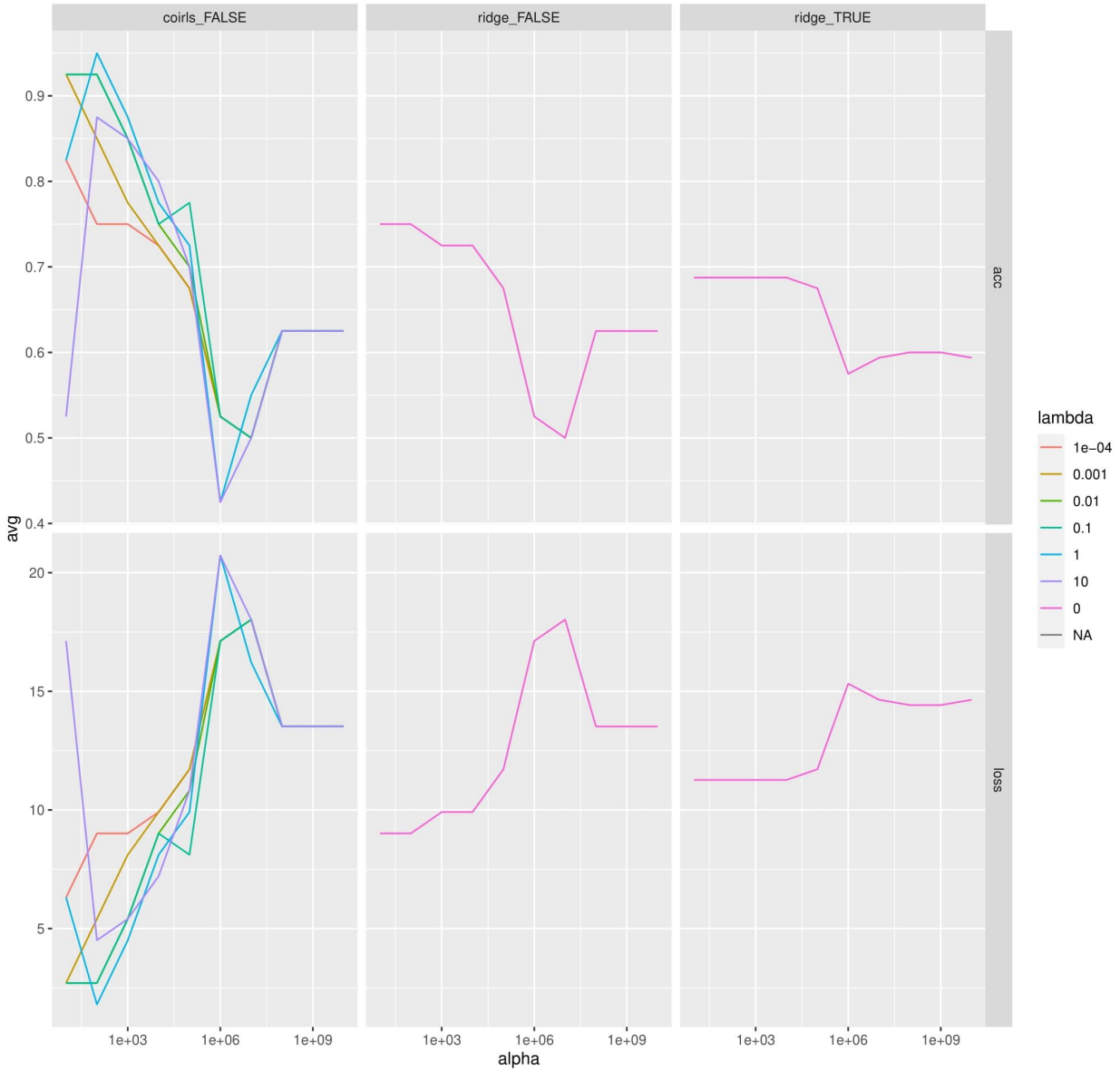
sub−17304


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.75
0.70
0.65
0.60
0.55
16
14
12
10
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

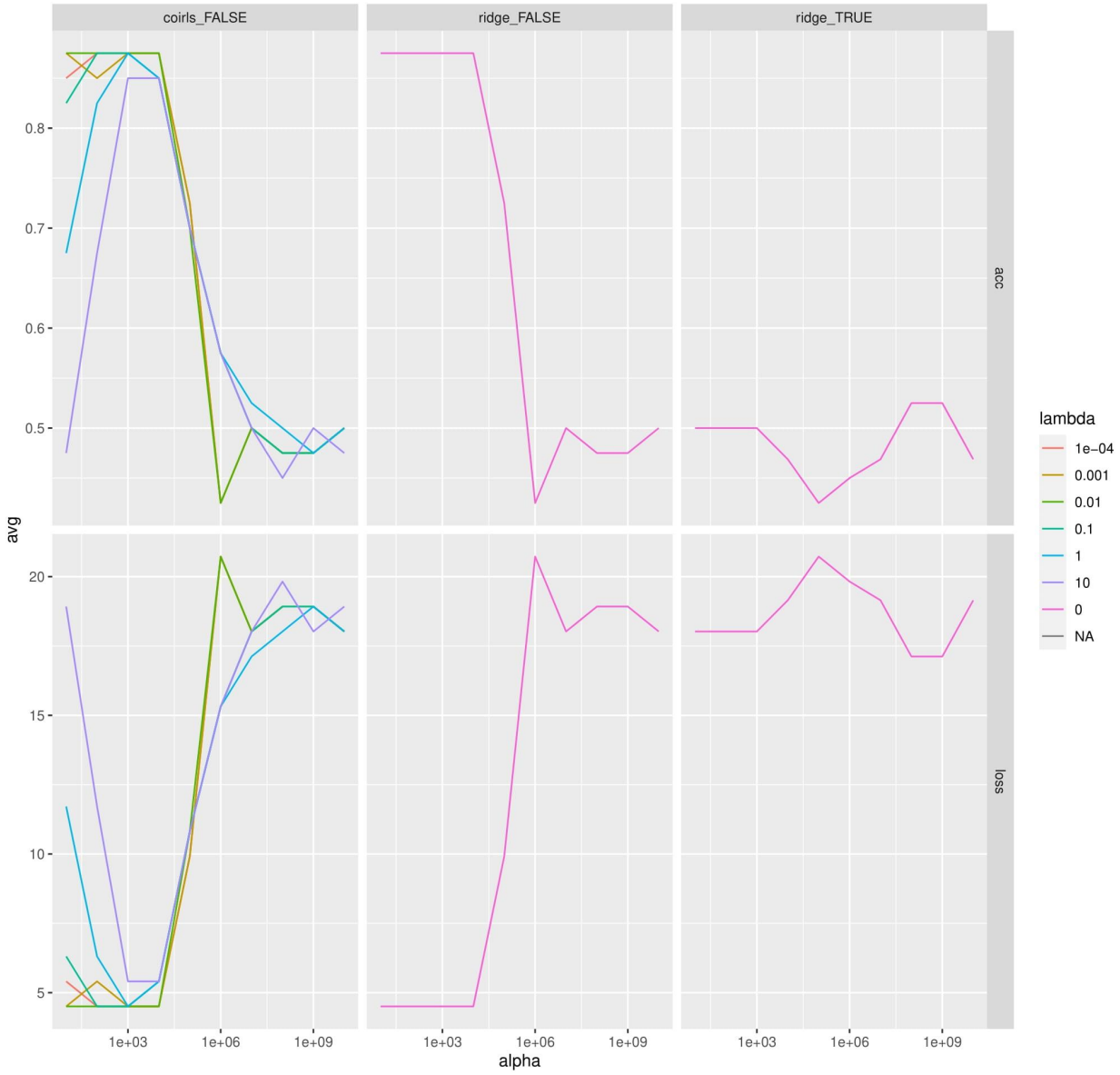
sub−17305


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.7
0.6
0.5
0.4
21
18
15
12
9
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

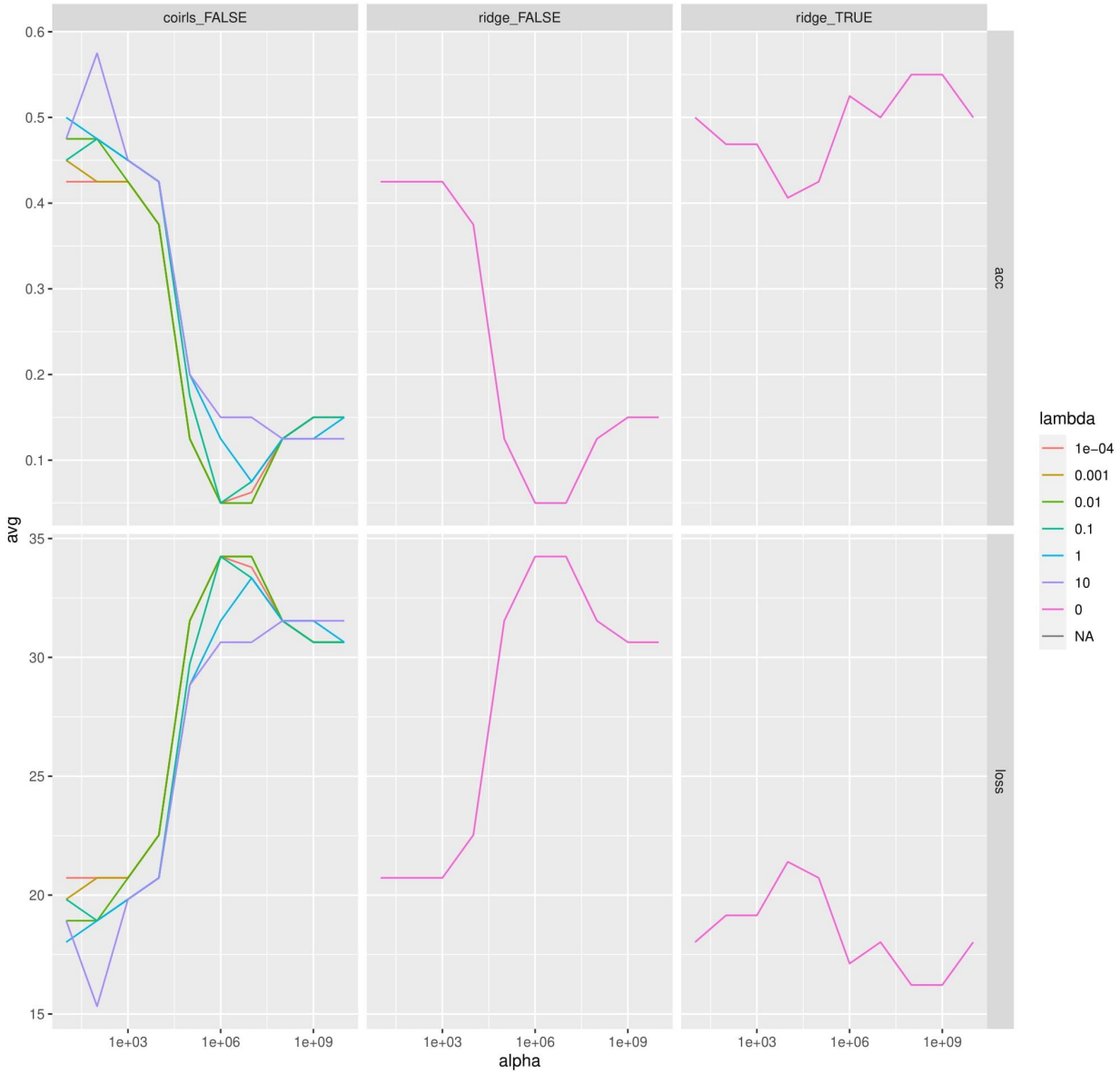
sub−17312


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.9
0.8
0.7
0.6
0.5
0.4
20
15
10
5
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

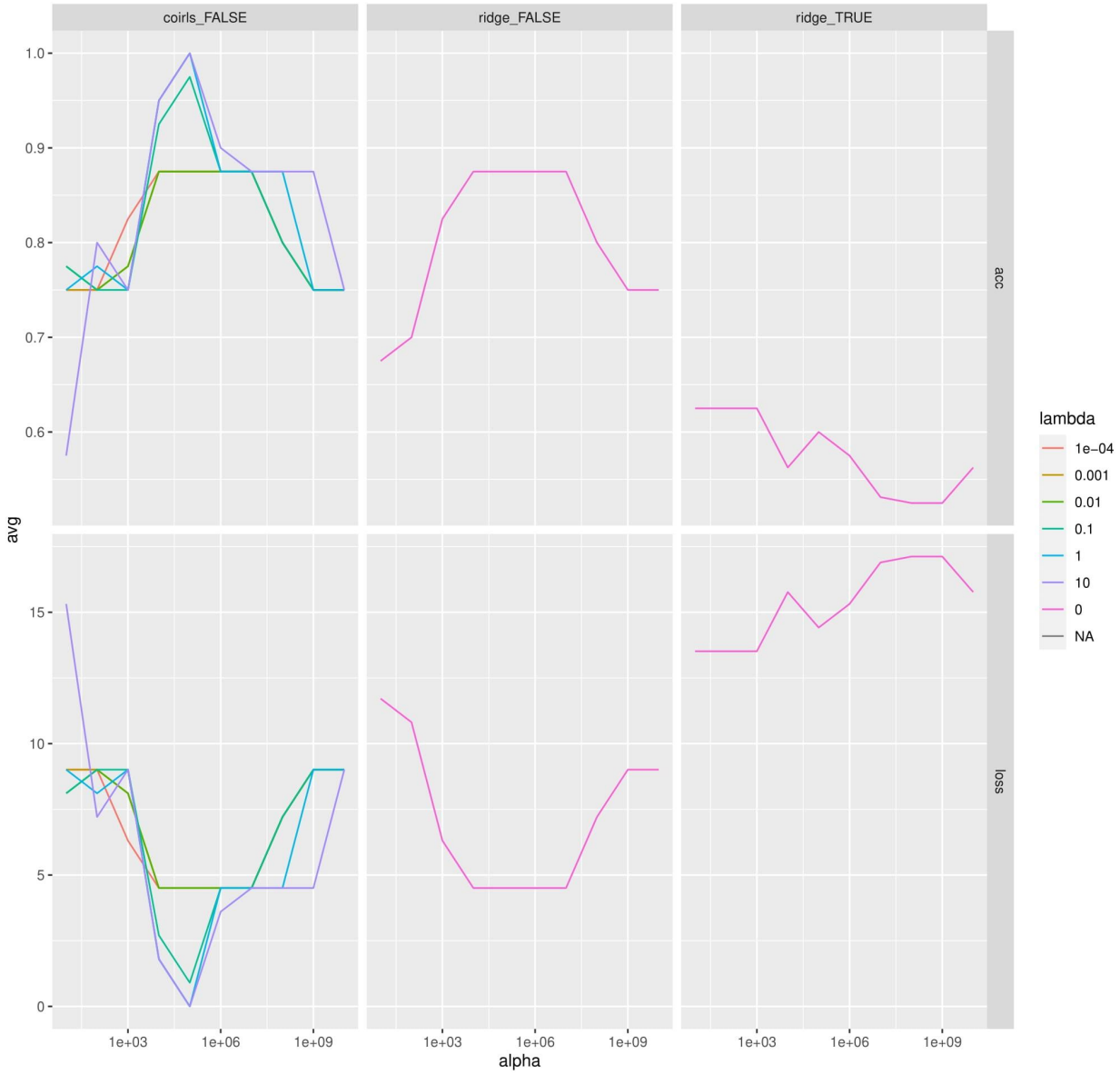
sub−17313


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.8
0.7
0.6
0.5
20
15
10
5
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

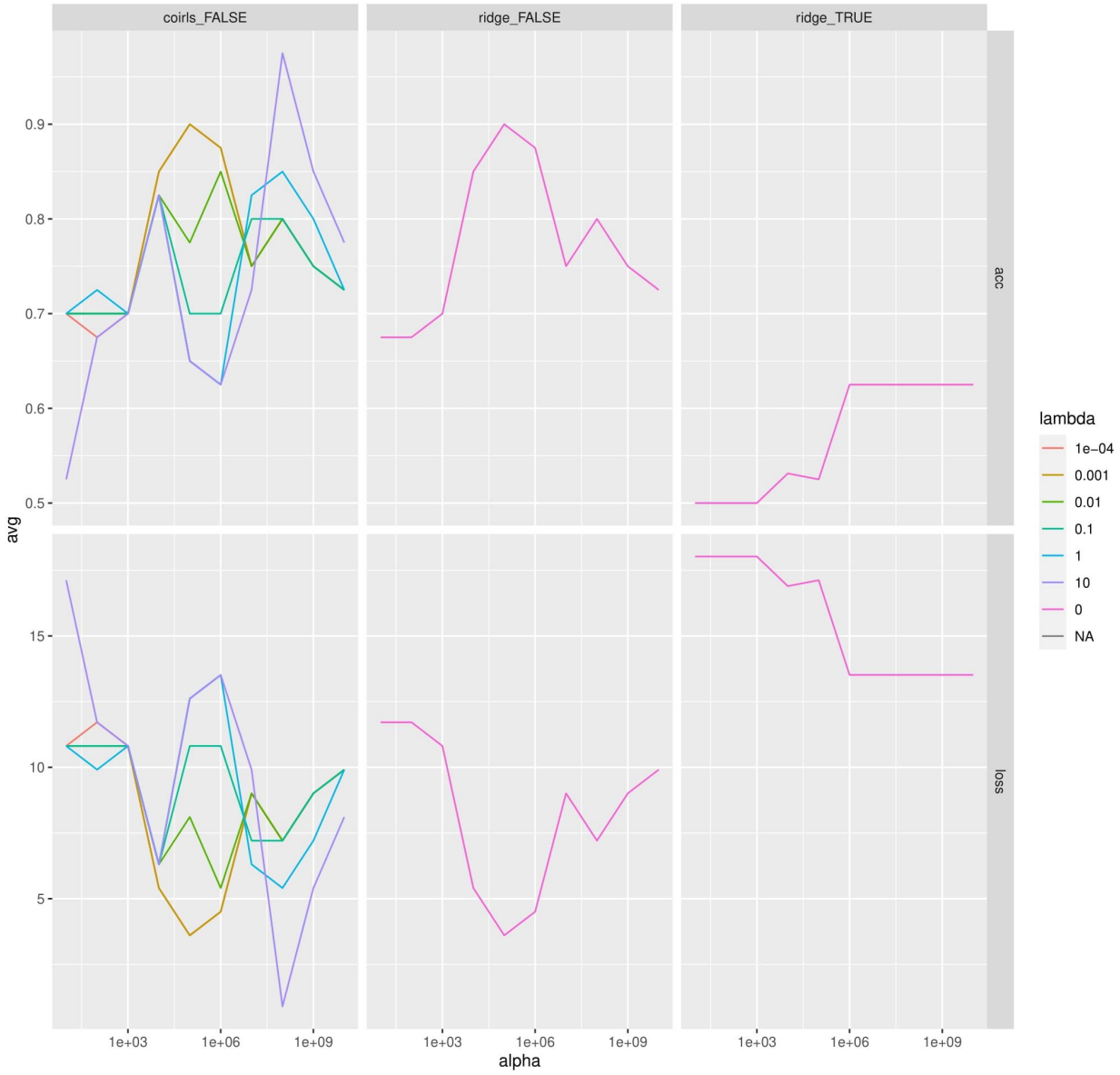
sub−17314


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.6
0.5
0.4
0.3
0.2
0.1
35
30
25
20
15
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

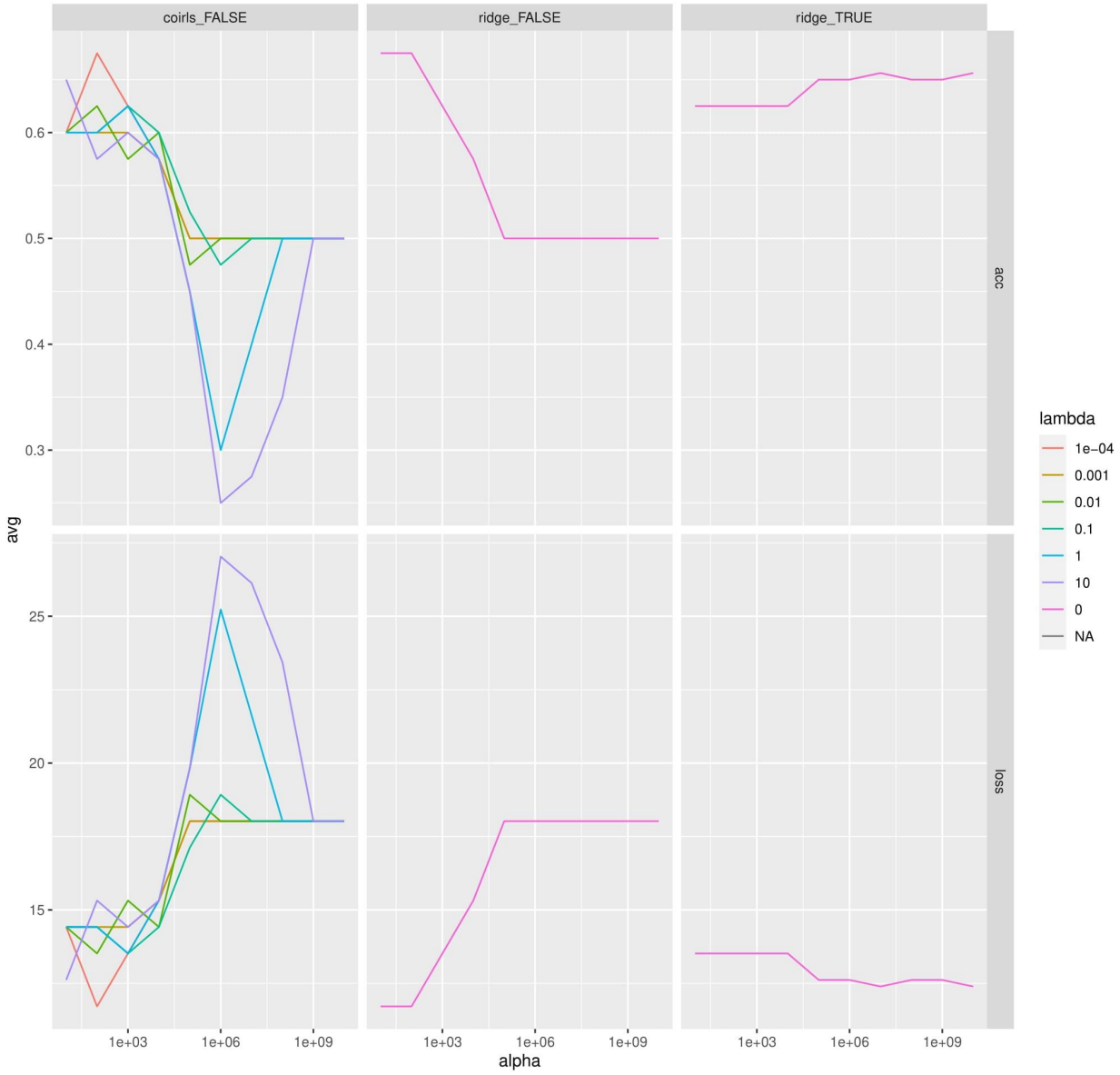
sub−17315


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
1.0
0.9
0.8
0.7
0.6
15
10
5
0
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

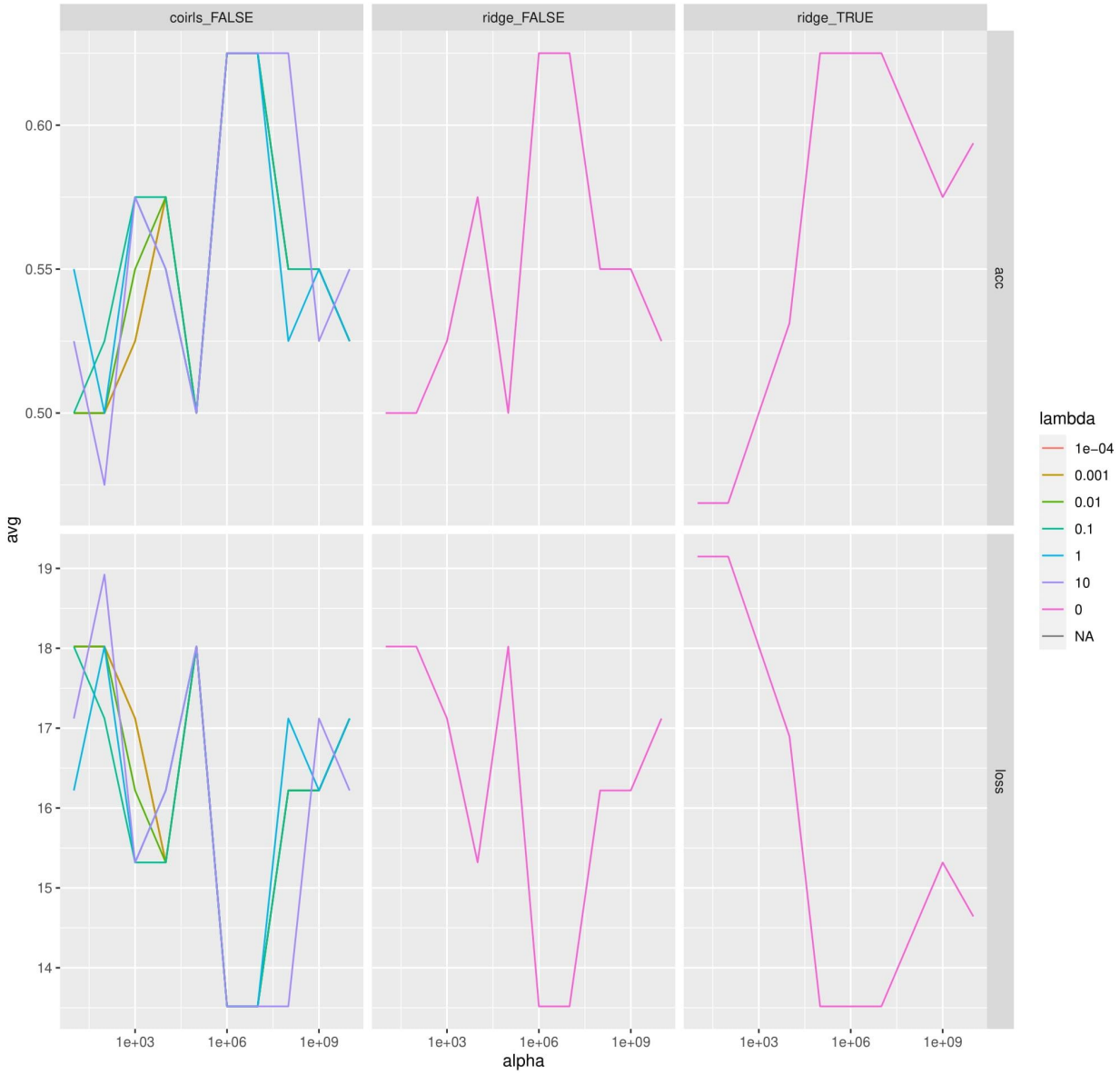
sub−17316

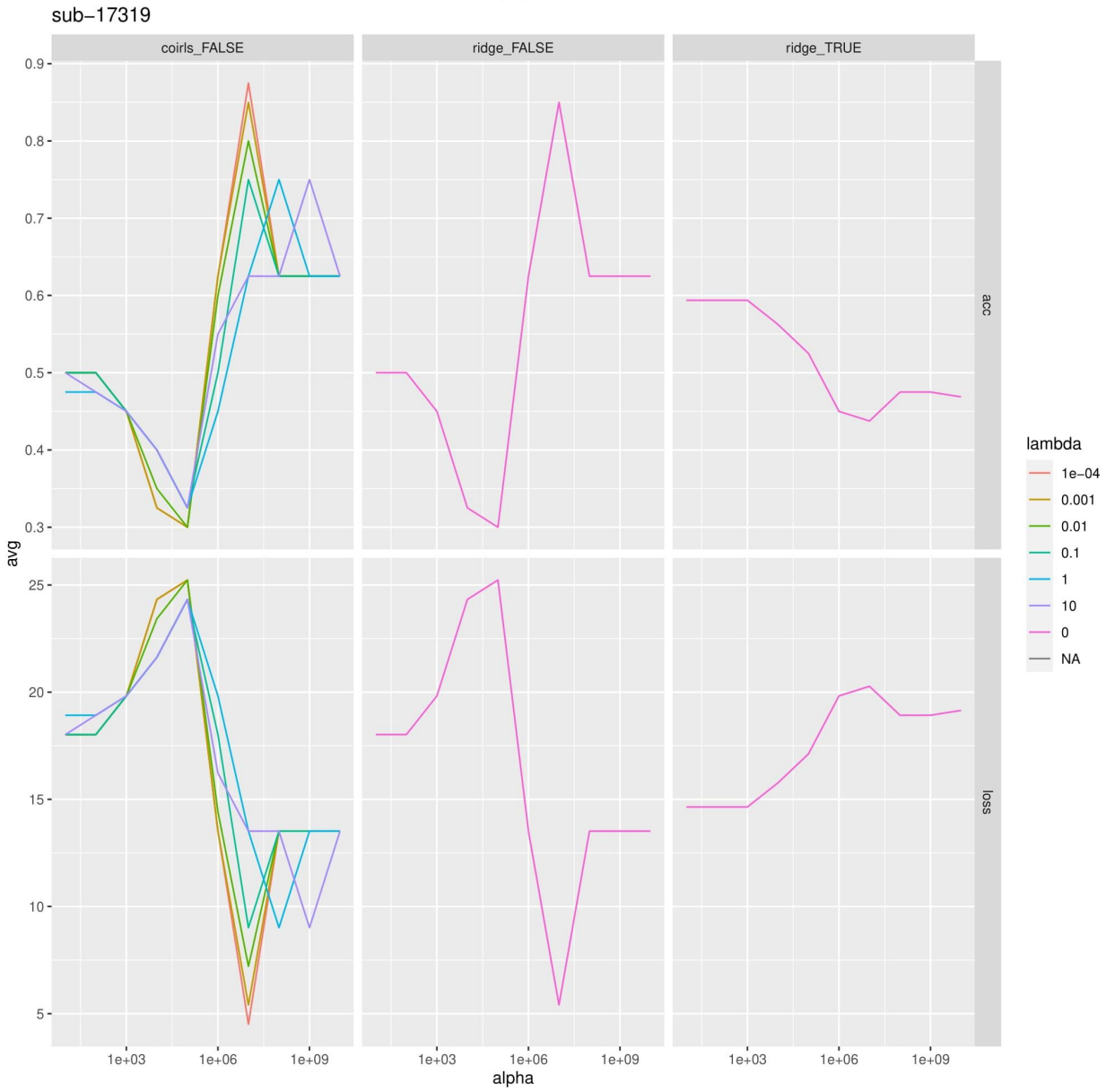

coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.9
0.8
0.7
0.6
0.5
15
10
5
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

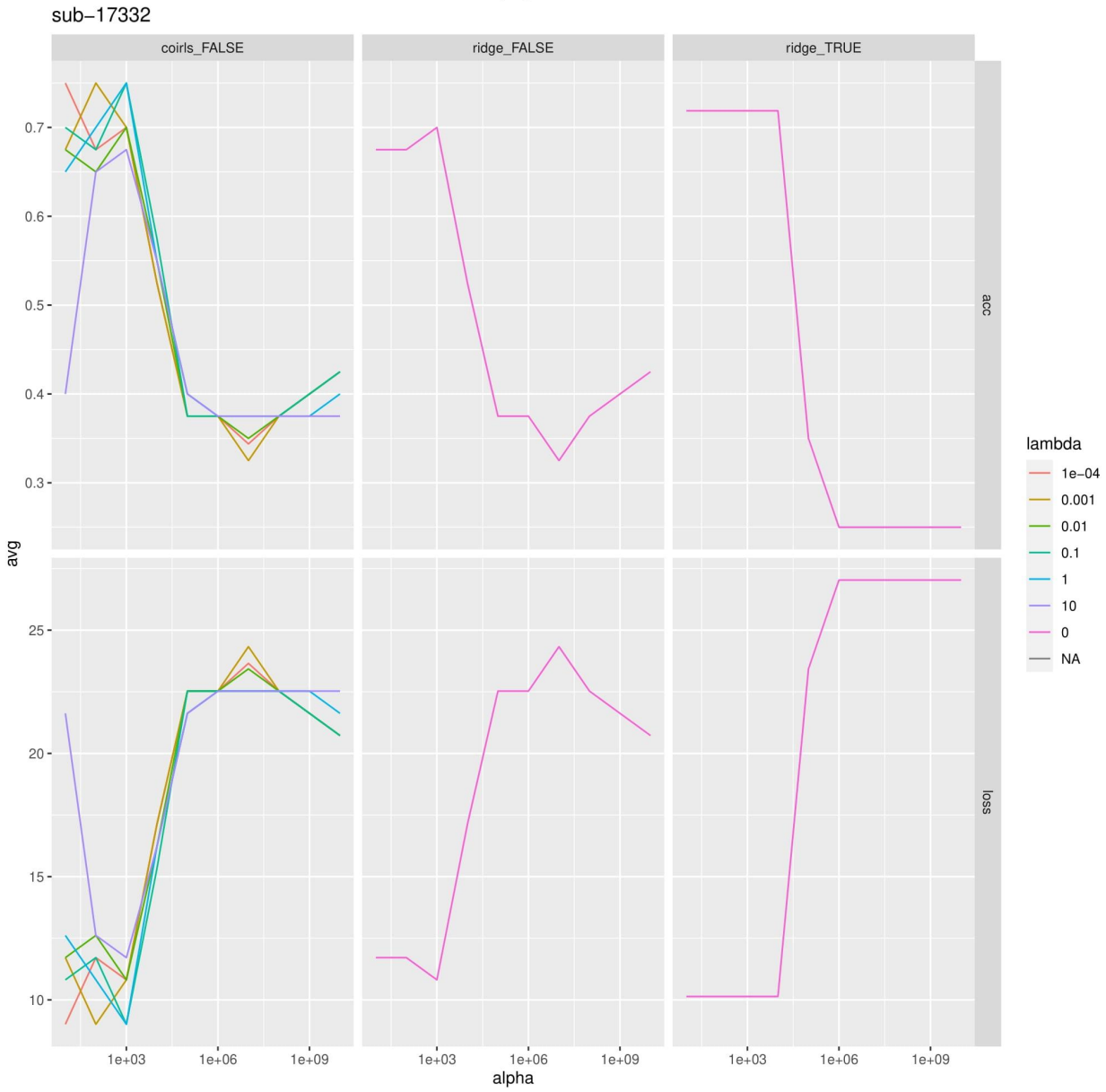
sub−17317


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.6
0.5
0.4
0.3
25
20
15
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

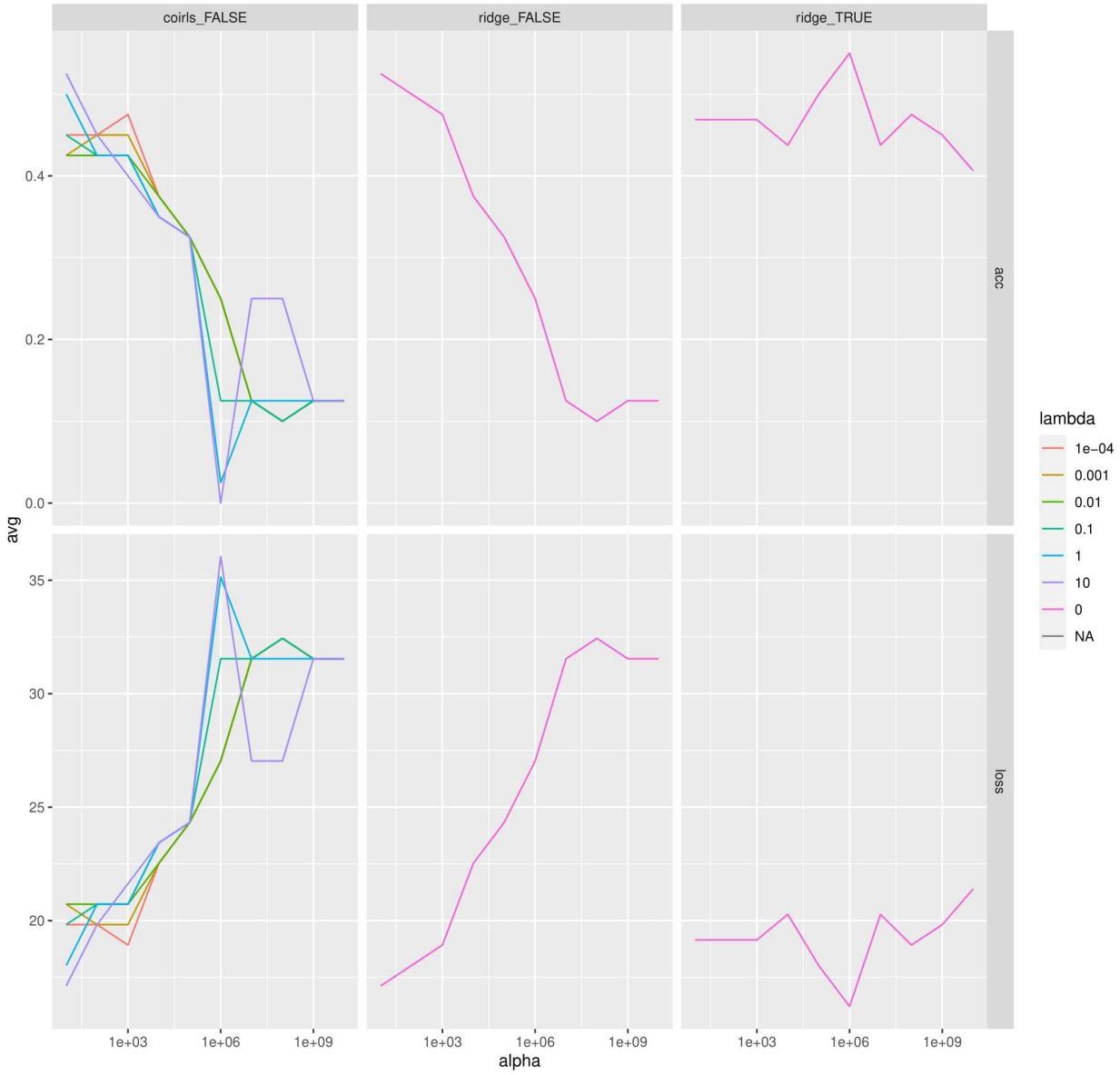
sub−17318

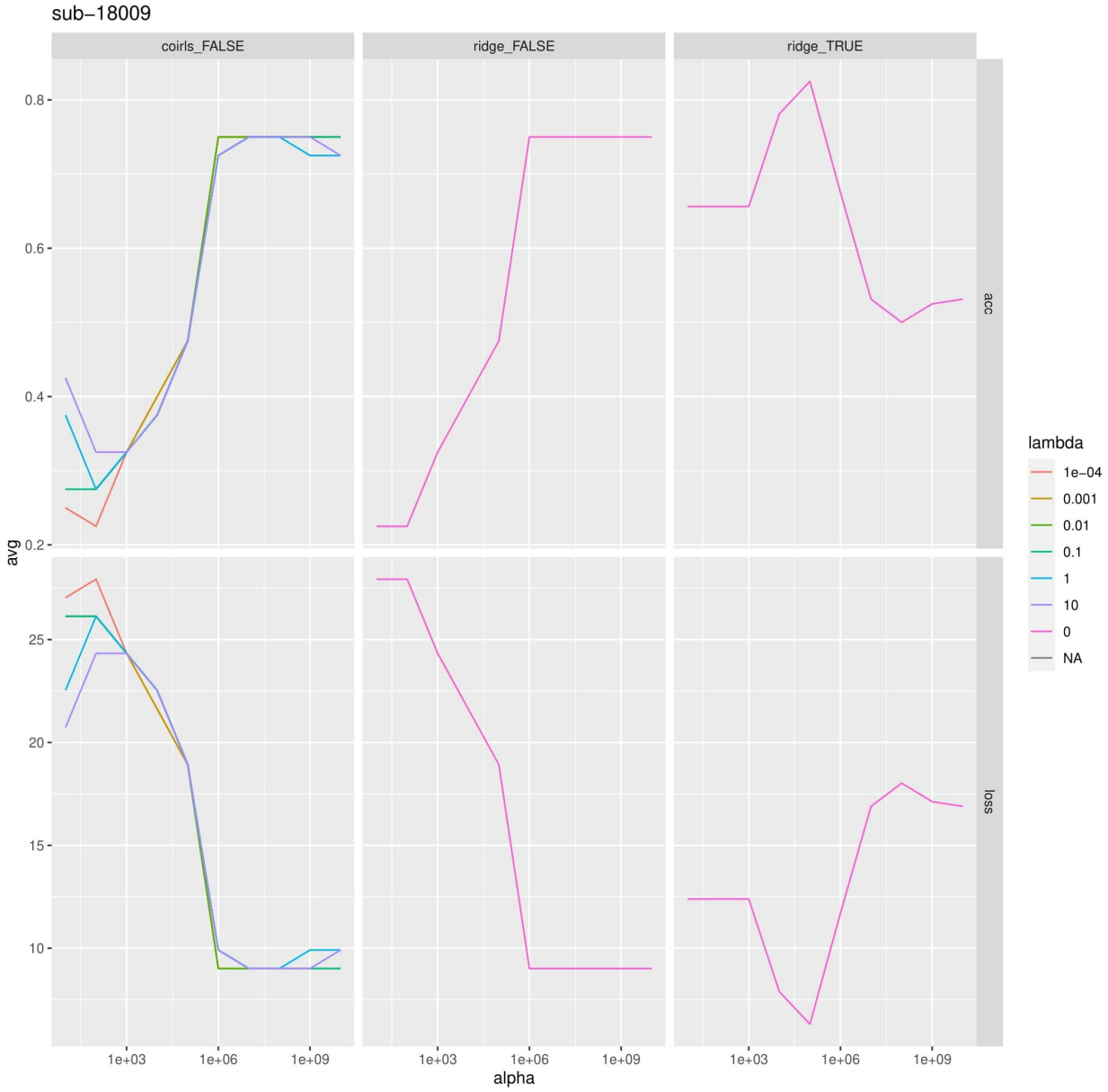

coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
0.60
0.55
0.50
19
18
17
16
15
14
1e+03
1e+06
1e+09
alpha
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA


sub−17319
coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.9
0.8
0.7
0.6
0.5
0.4
0.3
25
20
15
10
5
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

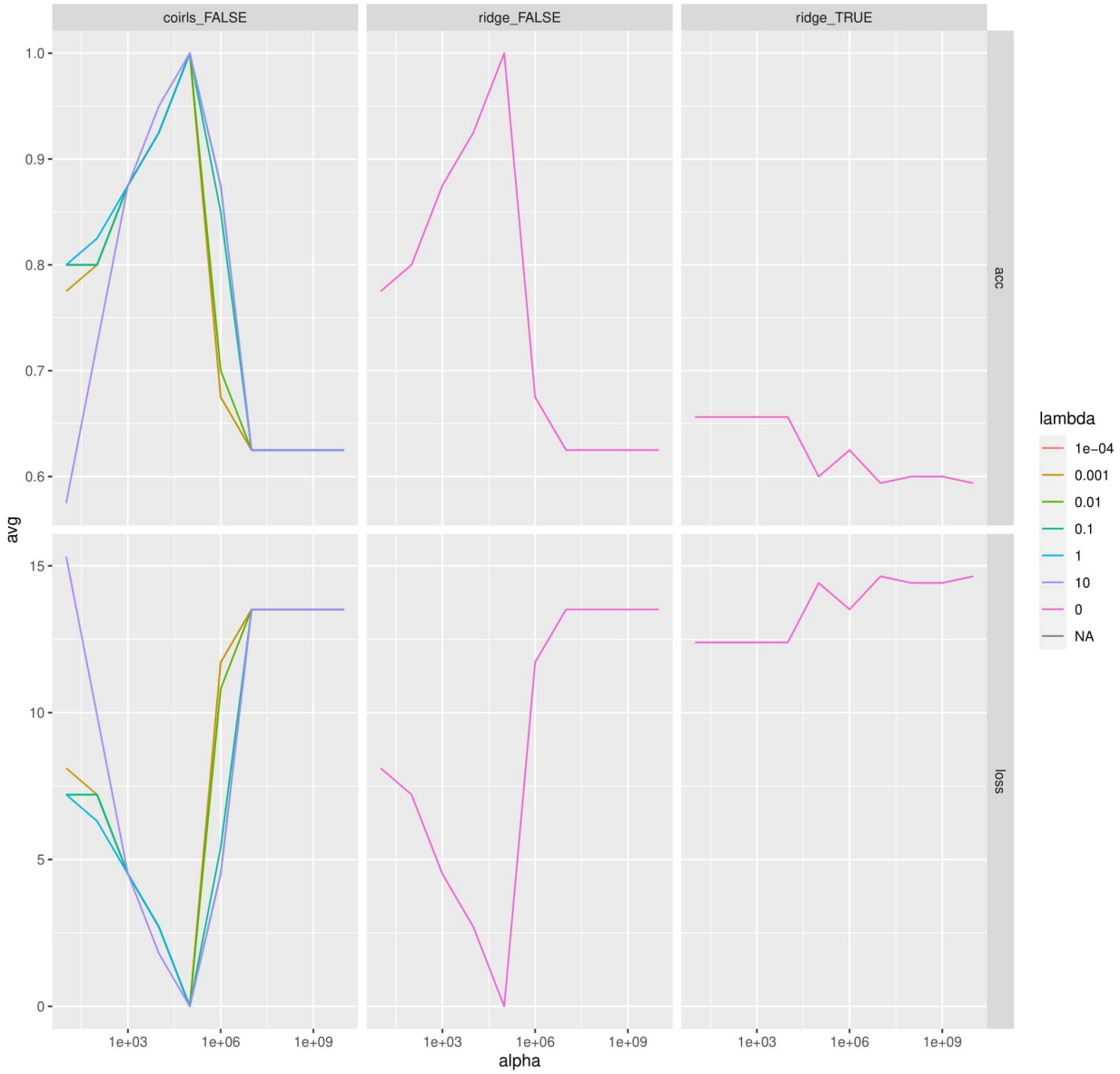

sub−17332
coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.7
0.6
0.5
0.4
0.3
25
20
15
10
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

sub−17333

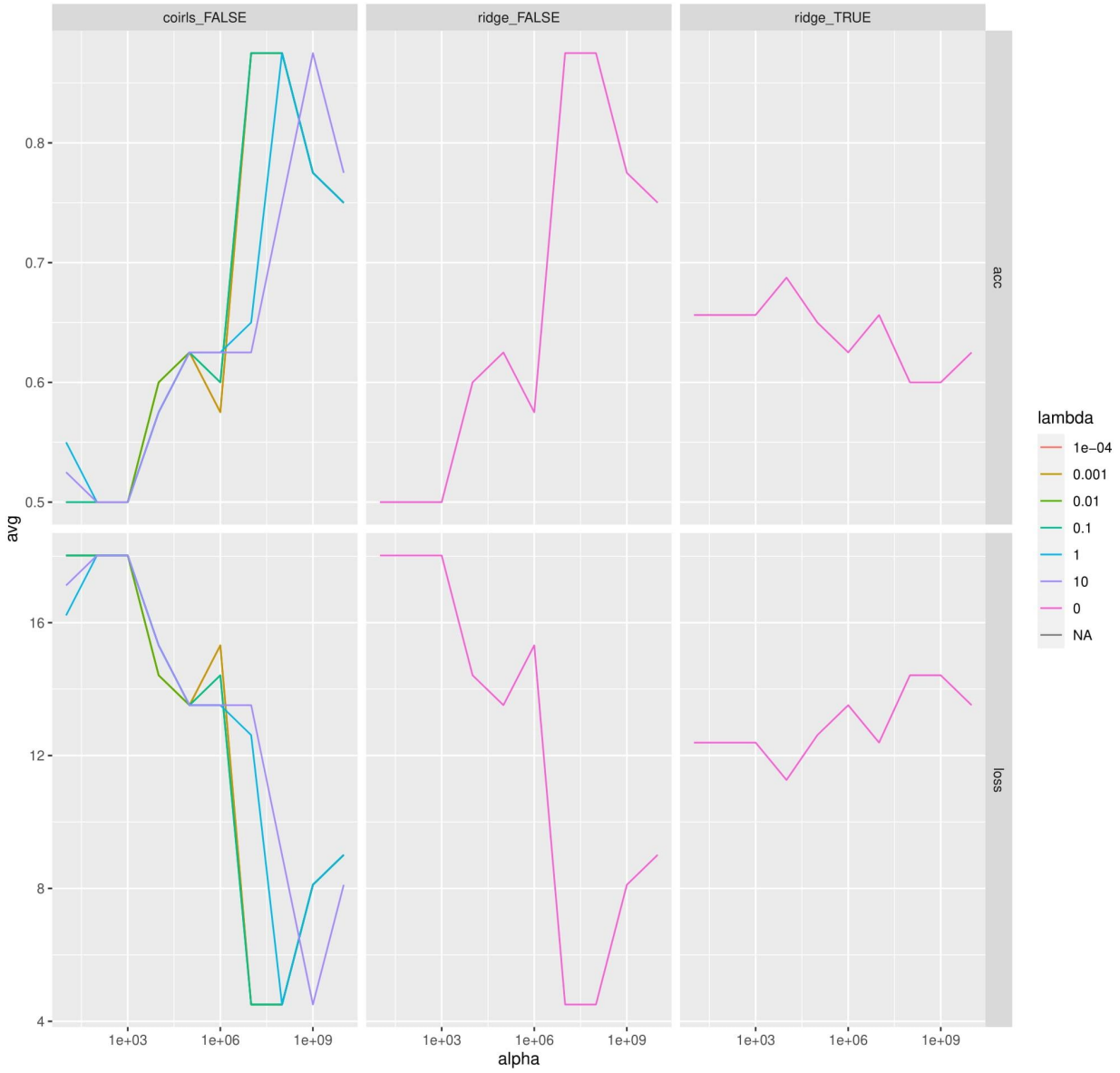

coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.0
0.2
0.4
20
25
30
35
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA


sub−18009
coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.8
0.6
0.4
0.2
25
20
15
10
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

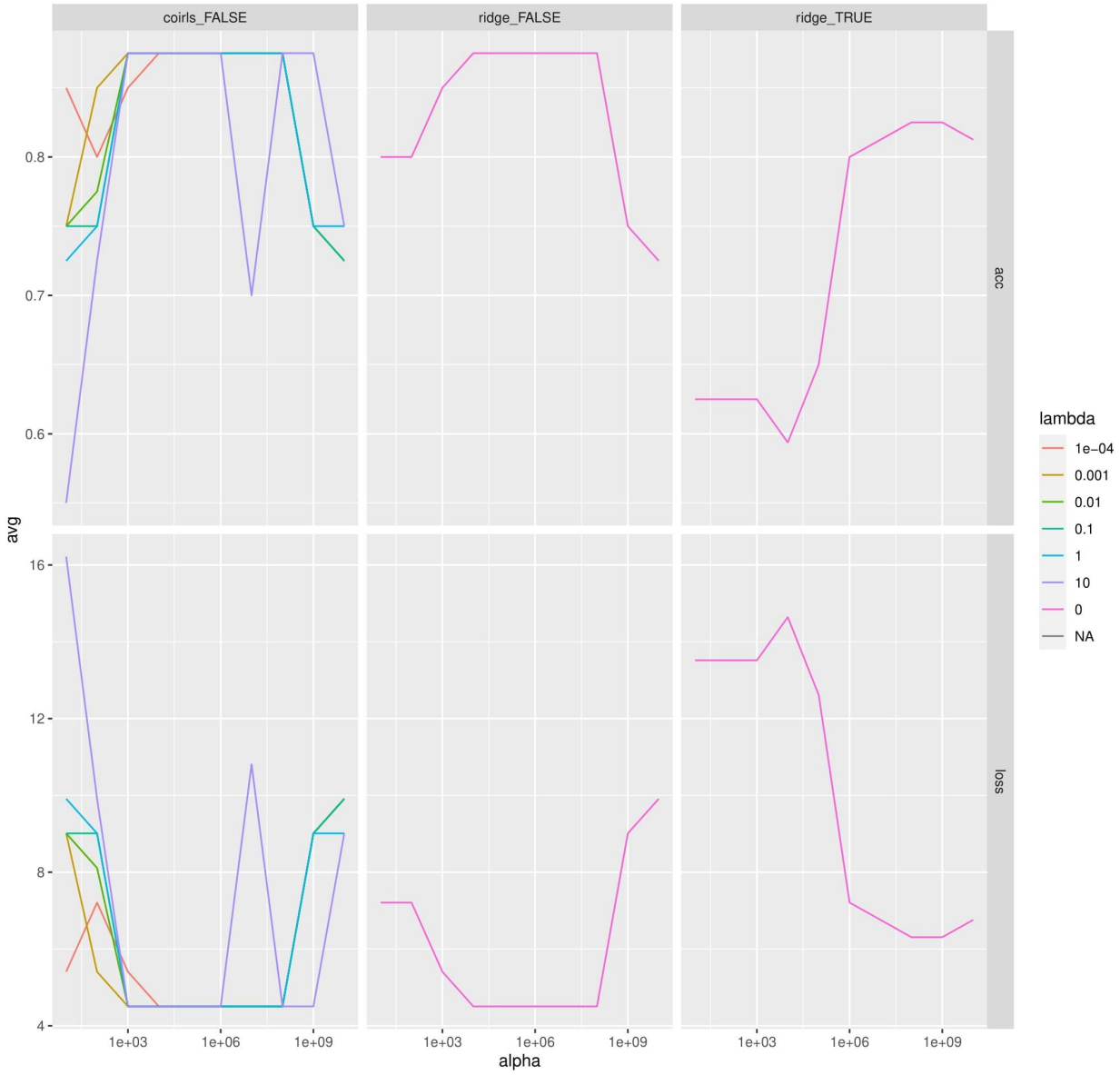
sub−18072

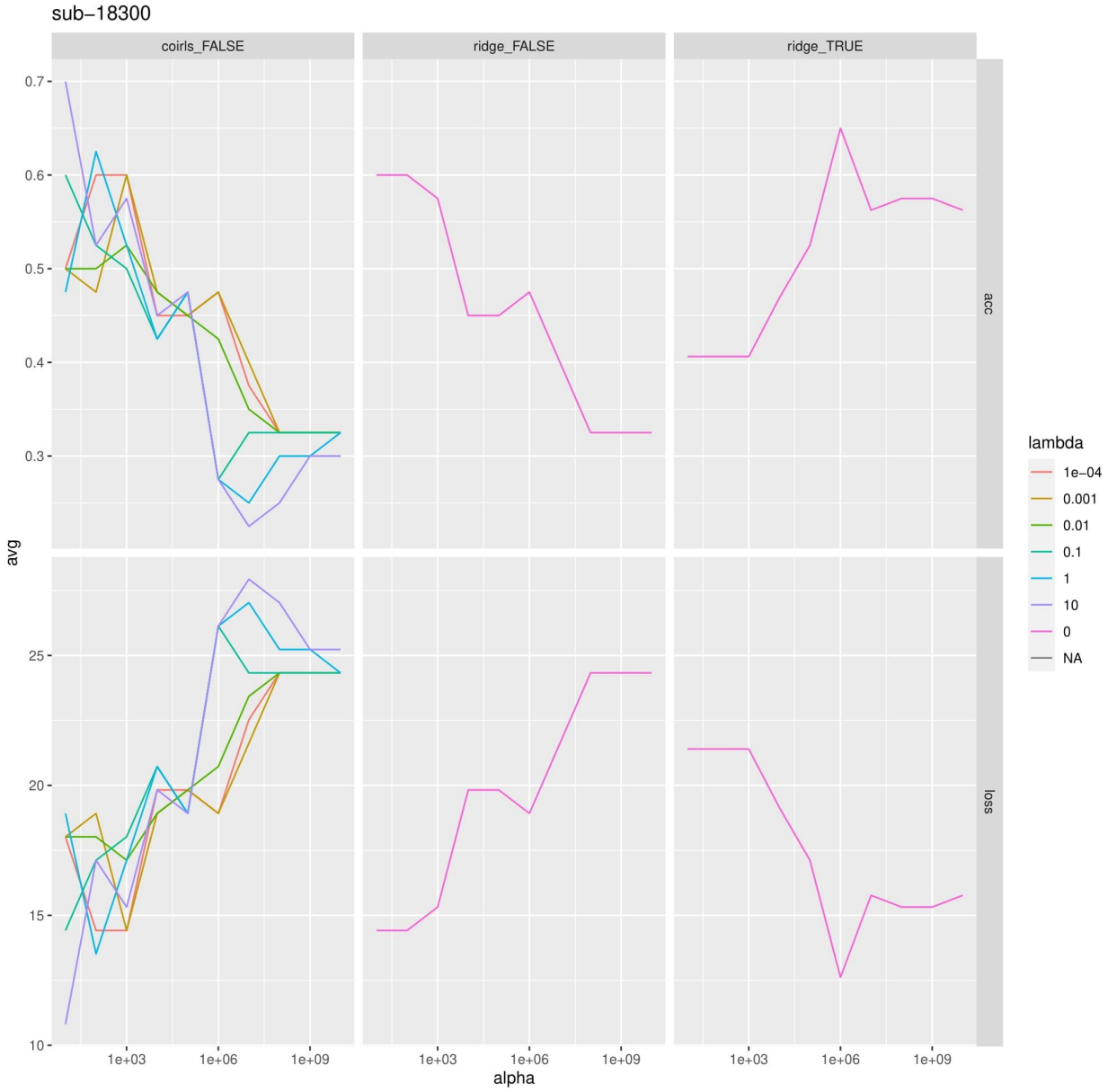

coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
1.0
0.9
0.8
0.7
0.6
15
10
5
0
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

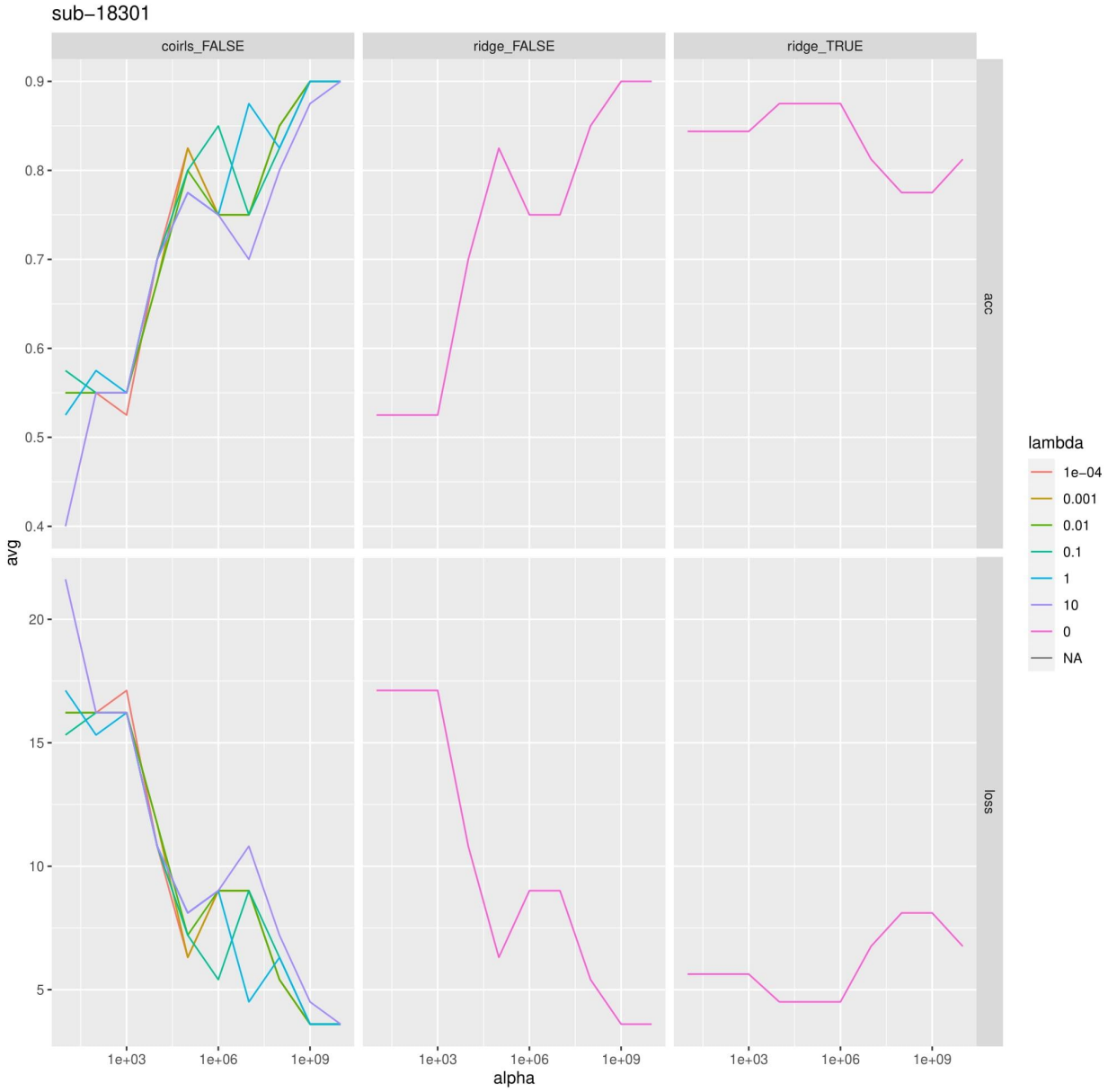
sub−18294


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.8
0.7
0.6
0.5
16
12
8
4
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

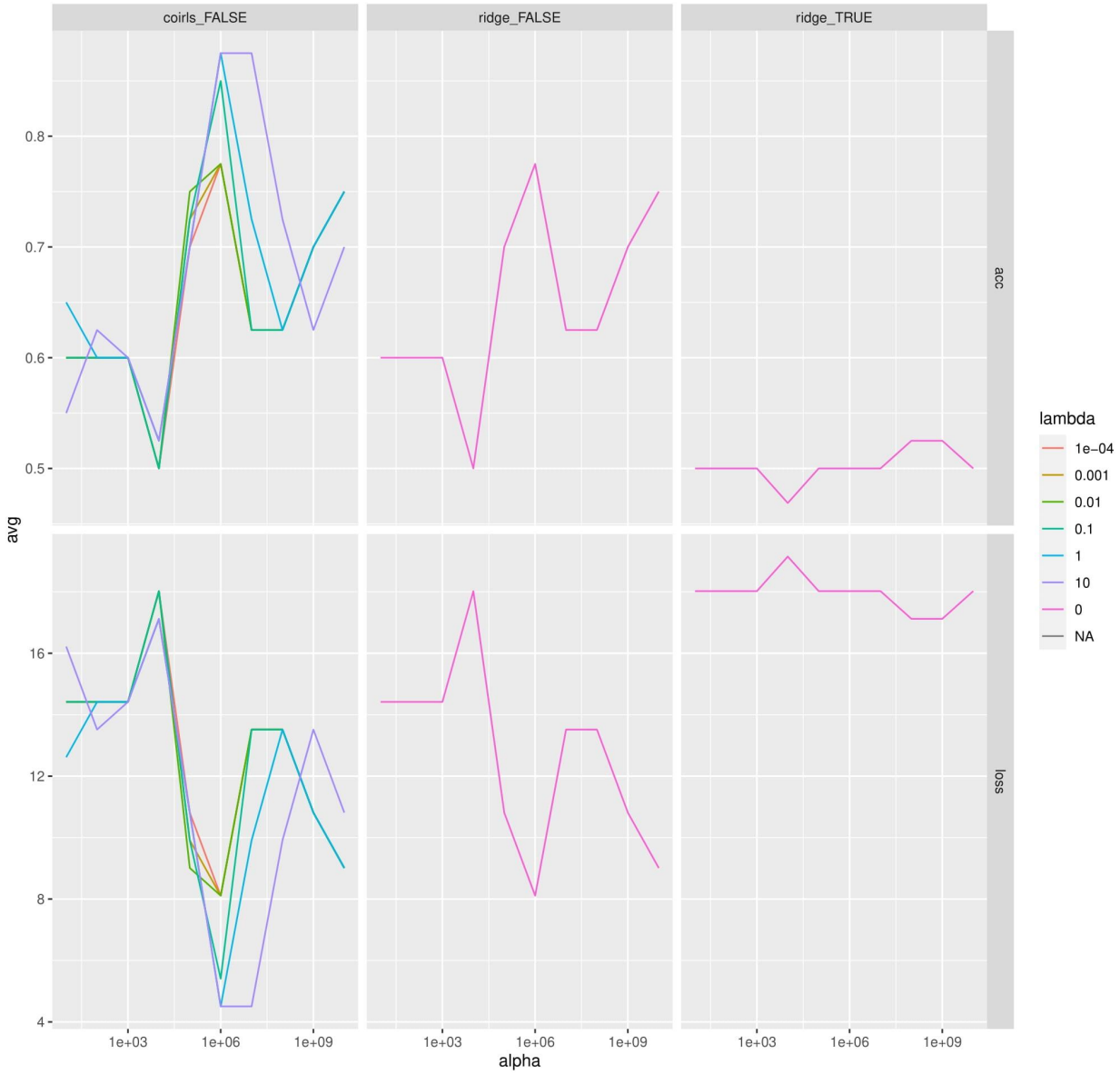
sub−18295


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.8
0.7
0.6
16
12
8
4
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

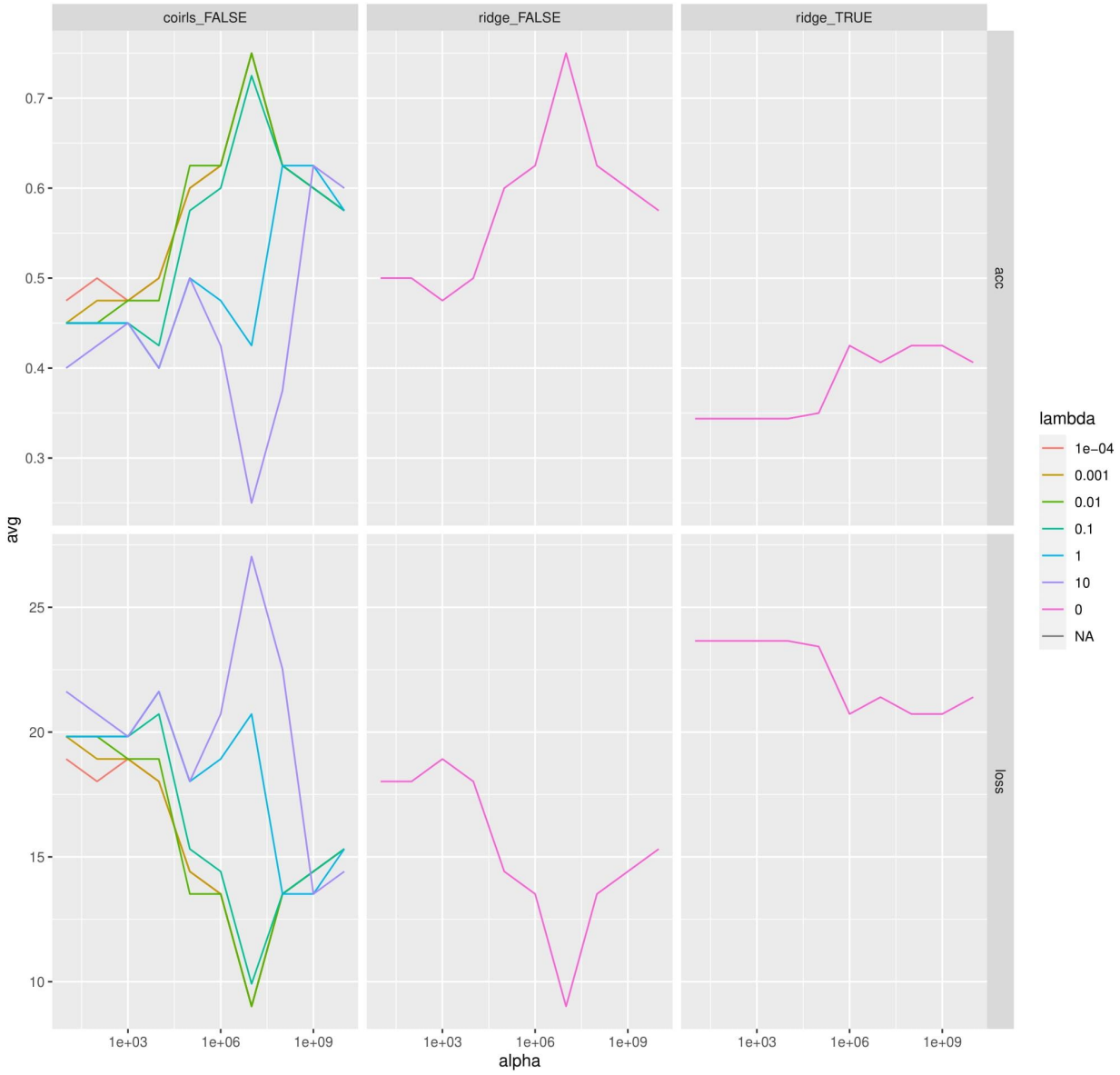

sub−18300
coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.7
0.6
0.5
0.4
0.3
25
20
15
10
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA


sub−18301
coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.9
0.8
0.7
0.6
0.5
0.4
20
15
10
5
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

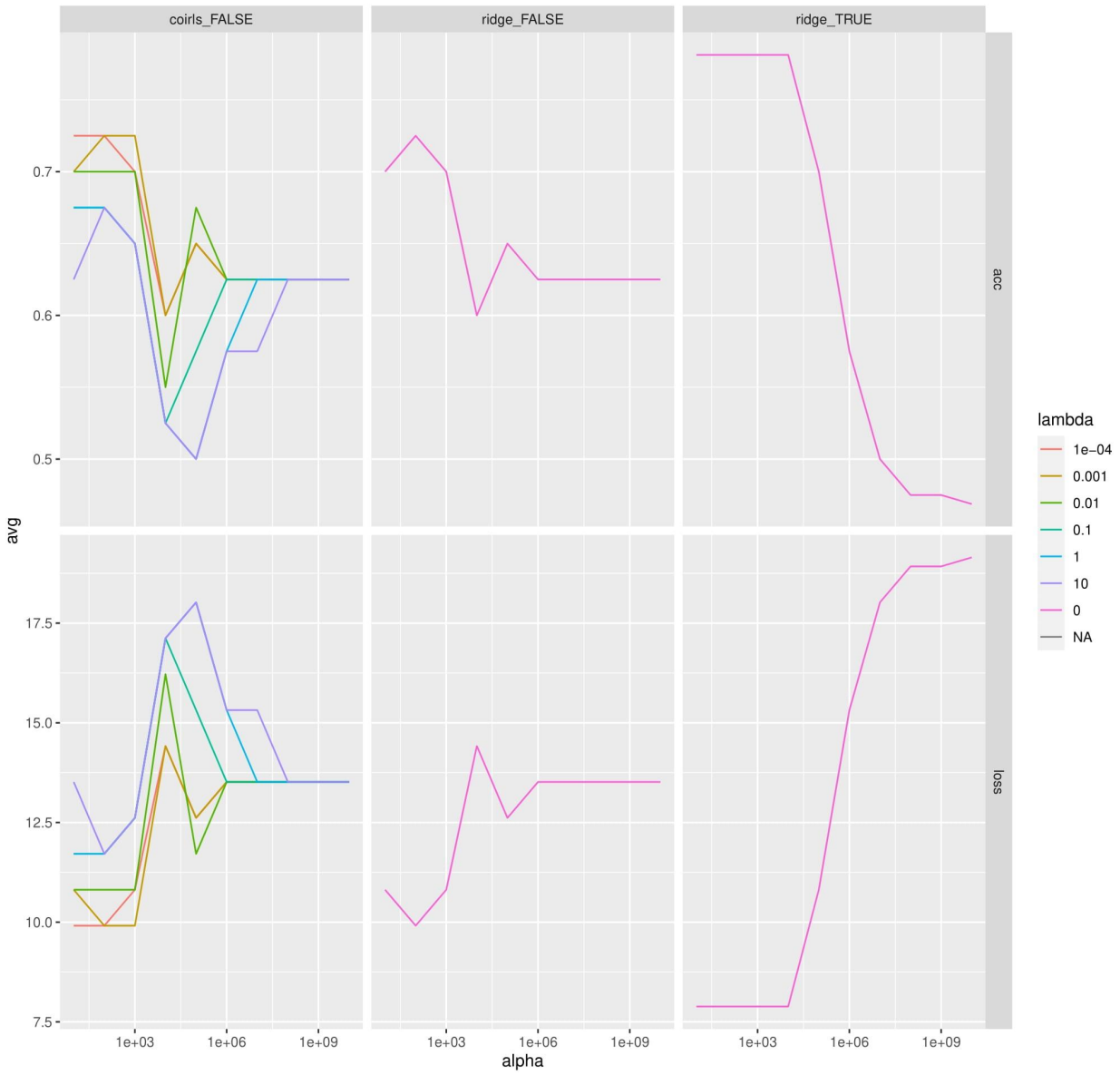
sub−18305


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.8
0.7
0.6
0.5
16
12
8
4
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

sub−18318


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.7
0.6
0.5
0.4
0.3
25
20
15
10
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

sub−18319


coirls_FALSE
ridge_FALSE
ridge_TRUE
acc
loss
avg
alpha
0.7
0.6
0.5
17.5
15.0
12.5
10.0
7.5
1e+03
1e+06
1e+09
lambda
1e−04
0.001
0.01
0.1
1
10
0
NA

Bibliography

Agren, T., Björkstrand, J., & Fredrikson, M. (2017). Disruption of human fear reconsolidation using imaginal and in vivo extinction. *Behavioural Brain Research*, *319*, 9–15. https://doi.org/10.1016/j.bbr.2016.11.014

Albers, A. M., Kok, P., Toni, I., Dijkerman, H. C., & de Lange, F. P. (2013). Shared representations for working memory and mental imagery in early visual cortex. *Current Biology*, *23*(15), 1427–1431. https://doi.org/10.1016/j.cub.2013.05.065

Alzubi, J., Nayyar, A., & Kumar, A. (2018). Machine Learning from Theory to Algorithms: An Overview. *Journal of Physics: Conference Series*, *1142*, 012012. https://doi.org/10.1088/1742-6596/1142/1/012012

Antoniadis, E. A., Winslow, J. T., Davis, M., & Amaral, D. G. (2009). The nonhuman primate amygdala is necessary for the acquisition but not the retention of fear-potentiated startle. *Biological Psychiatry*, *65*(3), 241–248. https://doi.org/10.1016/j.biopsych.2008.07.007

Arntz, A., Tiesema, M., & Kindt, M. (2007). Treatment of PTSD: a comparison of imaginal exposure with and without imagery rescripting. *Journal of Behavior Therapy and Experimental Psychiatry*, *38*(4), 345–370. https://doi.org/10.1016/j.jbtep.2007.10.006

Bellman, R. (1961).Adaptive Control Processes; A Guided Tour, Princeton Univ. *Press, NJ.*

Burleigh, L., & Greening, S. G. (2023). Fear in the mind's eye: The neural correlates of differential fear acquisition to imagined conditioned stimuli. *Social Cognitive and Affective Neuroscience*.

Burleigh, L., Jiang, X., & Greening, S. G. (2022). Fear in the theater of the mind: differential fear conditioning with imagined stimuli. *Psychological Science*, 095679762210865. https://doi.org/10.1177/09567976221086513

Carter, C. S., Perlstein, W., Ganguli, R., Brar, J., Mintun, M., & Cohen, J. D. (1998). Functional hypofrontality and working memory dysfunction in schizophrenia. *The American Journal of Psychiatry*, *155*(9), 1285–1287. https://doi.org/10.1176/ajp.155.9.1285

Clasen, M. (2012). Monsters evolve: A biocultural approach to horror stories. *Review of General Psychology*, *16*(2), 222–229. https://doi.org/10.1037/a0027918

D'Esposito, M., Deouell, L. Y., & Gazzaley, A. (2003). Alterations in the BOLD fMRI signal with ageing and disease: a challenge for neuroimaging. *Nature Reviews. Neuroscience*, *4*(11), 863–872. https://doi.org/10.1038/nrn1246

Dadds, M. R., Bovbjerg, D. H., Redd, W. H., & Cutmore, T. R. (1997). Imagery in human classical conditioning. *Psychological Bulletin*, *122*(1), 89–103. https://doi.org/10.1037/0033-2909.122.1.89

Dale, A. M. (1999). Optimal experimental design for event-related fMRI. *Human Brain Mapping*, *8*(2–3), 109–114.

https://doi.org/10.1002/(SICI)1097-0193(1999)8:2/3<109::AID-HBM7>3.0.CO;2-W

Di Bono, M., & Zorzi, M. (2008). Decoding Cognitive States from fMRI Data Using Support Vector Regression. *PsychNology*.

Dunsmoor, J. E., Martin, A., & LaBar, K. S. (2012). Role of conceptual knowledge in learning and retention of conditioned fear. *Biological Psychology*, *89*(2), 300–305. https://doi.org/10.1016/j.biopsycho.2011.11.002

Fullana, M. A., Harrison, B. J., Soriano-Mas, C., Vervliet, B., Cardoner, N., Àvila-Parcet, A., & Radua, J. (2016). Neural signatures of human fear conditioning: an updated and extended meta-analysis of fMRI studies. *Molecular Psychiatry*, *21*(4), 500–508. https://doi.org/10.1038/mp.2015.88

Ganis, G., Thompson, W. L., & Kosslyn, S. M. (2004). Brain areas underlying visual mental imagery and visual perception: an fMRI study. *Brain Research. Cognitive Brain Research*, *20*(2), 226–241. https://doi.org/10.1016/j.cogbrainres.2004.02.012

Greening, S. G., Lee, T.-H., Grégoire, L., Burleigh, L., Robinson, T., Jiang, X., Mather, M., & Kaplan, J. (2021). Fear in the mind's eye: Mental imagery can generate and regulate acquired differential fear conditioned reactivity. *BioRxiv*. https://doi.org/10.1101/2021.02.04.429795

Greening, S. G., Lee, T.-H., & Mather, M. (2016). Individual Differences in Anticipatory Somatosensory Cortex Activity for Shock is Positively Related with Trait Anxiety and Multisensory Integration. *Brain Sciences*, *6*(1). https://doi.org/10.3390/brainsci6010002

Grégoire, L., & Greening, S. G. (2019). Opening the reconsolidation window using the mind's eye: Extinction training during reconsolidation disrupts fear memory expression following mental imagery reactivation. *Cognition*, *183*, 277–281. https://doi.org/10.1016/j.cognition.2018.12.001

Grégoire, L., & Greening, S. G. (2020). Fear of the known: semantic generalisation of fear conditioning across languages in bilinguals. *Cognition & Emotion*, *34*(2), 352–358. https://doi.org/10.1080/02699931.2019.1604319

Grunert, B. K., Weis, J. M., Smucker, M. R., & Christianson, H. F. (2007). Imagery rescripting and reprocessing therapy after failed prolonged exposure for post-traumatic stress disorder following industrial injury. *Journal of Behavior Therapy and Experimental Psychiatry*, *38*(4), 317–328. https://doi.org/10.1016/j.jbtep.2007.10.005

Gui, L., Xu, R., Lu, Q., Du, J., & Zhou, Y. (2017). Negative transfer detection in transductive transfer learning. *International Journal of Machine Learning and Cybernetics*, *9*(2), 185–197. https://doi.org/10.1007/s13042-016-0634-8

Gullick, M. M., Mitra, P., & Coch, D. (2013). Imagining the truth and the moon: an electrophysiological study of abstract and concrete word processing. *Psychophysiology*, *50*(5), 431–440. https://doi.org/10.1111/psyp.12033

Hackmann, A., & Holmes, E. A. (2004). Reflecting on imagery: a clinical perspective and

overview of the special issue of memory on mental imagery and memory in psychopathology. *Memory*, *12*(4), 389–402. https://doi.org/10.1080/09658210444000133

Handwerker, D. A., Ollinger, J. M., & D'Esposito, M. (2004). Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses. *Neuroimage*, *21*(4), 1639–1651. https://doi.org/10.1016/j.neuroimage.2003.11.029

Hanke, M., Halchenko, Y. O., Sederberg, P. B., Hanson, S. J., Haxby, J. V., & Pollmann, S. (2009). PyMVPA: A python toolbox for multivariate pattern analysis of fMRI data. *Neuroinformatics*, *7*(1), 37–53. https://doi.org/10.1007/s12021-008-9041-y

Haruno, M., & Kawato, M. (2006). Heterarchical reinforcement-learning model for integration of multiple cortico-striatal loops: fMRI examination in stimulus-action-reward association learning. *Neural Networks*, *19*(8), 1242–1254. https://doi.org/10.1016/j.neunet.2006.06.007

Hendriks, M. H. A., Daniels, N., Pegado, F., & Op de Beeck, H. P. (2017). The effect of spatial smoothing on representational similarity in a simple motor paradigm. *Frontiers in Neurology*, *8*, 222. https://doi.org/10.3389/fneur.2017.00222

Holmes, E. A., Arntz, A., & Smucker, M. R. (2007). Imagery rescripting in cognitive behaviour therapy: images, treatment techniques and outcomes. *Journal of Behavior Therapy and Experimental Psychiatry*, *38*(4), 297–305. https://doi.org/10.1016/j.jbtep.2007.10.007

Holmes, E. A., & Mathews, A. (2010). Mental imagery in emotion and emotional disorders. *Clinical Psychology Review*, *30*(3), 349–362. https://doi.org/10.1016/j.cpr.2010.01.001

Holzman, A. D., & Levis, D. J. (1991). Differential Aversive Conditioning of an External (Visual) and Internal (Imaginal) CS: Effects of Transfer Between and Within CS Modalities . *Journal of Mental Imagery*.

Hoppe, J. M., Holmes, E. A., & Agren, T. (2021). Exploring the neural basis of fear produced by mental imagery: imaginal exposure in individuals fearful of spiders. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *376*(1817), 20190690. https://doi.org/10.1098/rstb.2019.0690

Horowitz, M. J. (1969). Psychic trauma. Return of images after a stress film. *Archives of General Psychiatry*, *20*(5), 552–559. https://doi.org/10.1001/archpsyc.1969.01740170056008

Jiang, X., Burleigh, L., & Greening, S. G. (2021). Complete the triangulation: Quantifying differential fear conditioning with a noninterfering and sensitive behavioral measure along with self-report and physiological measures. *Psychophysiology*, e13831. https://doi.org/10.1111/psyp.13831

Jiang, X., & Greening, S. (2021). Psychophysiological evidence for fear extinction learning via mental imagery. *BioRxiv*. https://doi.org/10.1101/2021.06.17.448826

Ji, J. L., Heyes, S. B., MacLeod, C., & Holmes, E. A. (2016). Emotional Mental Imagery as Simulation of Reality: Fear and Beyond-A Tribute to Peter Lang. *Behavior Therapy*, *47*(5), 702–719. https://doi.org/10.1016/j.beth.2015.11.004

Kaplan, J. T., Man, K., & Greening, S. G. (2015). Multivariate cross-classification: applying machine learning techniques to characterize abstraction in neural representations. *Frontiers in Human Neuroscience*, *9*, 151. https://doi.org/10.3389/fnhum.2015.00151

Klein, I., Paradis, A. L., Poline, J. B., Kosslyn, S. M., & Le Bihan, D. (2000). Transient activity in the human calcarine cortex during visual-mental imagery: an event-related fMRI study. *Journal of Cognitive Neuroscience*, *12 Suppl 2*, 15–23. https://doi.org/10.1162/089892900564037

Knight, D. C., Nguyen, H. T., & Bandettini, P. A. (2005). The role of the human amygdala in the production of conditioned fear responses. *Neuroimage*, *26*(4), 1193–1200. https://doi.org/10.1016/j.neuroimage.2005.03.020

Koizumi, A., Amano, K., Cortese, A., Shibata, K., Yoshida, W., Seymour, B., Kawato, M., & Lau, H. (2016). Fear reduction without fear through reinforcement of neural activity that bypasses conscious exposure. *Nature Human Behaviour*, *1*. https://doi.org/10.1038/s41562-016-0006

Kosslyn, S. M., Ganis, G., & Thompson, W. L. (2001). Neural foundations of imagery. *Nature Reviews. Neuroscience*, *2*(9), 635–642. https://doi.org/10.1038/35090055

Kosslyn, S. M. (1988). Aspects of a cognitive neuroscience of mental imagery. *Science*, *240*(4859), 1621–1626. https://doi.org/10.1126/science.3289115

Kruggel, F., & von Cramon, D. Y. (1999). Temporal properties of the hemodynamic response in functional MRI. *Human Brain Mapping*, *8*(4), 259–271. https://doi.org/10.1002/(SICI)1097-0193(1999)8:4<259::AID-HBM9>3.0.CO;2-K

Krypotos, A.-M., Mertens, G., Leer, A., & Engelhard, I. M. (2020). Induction of conditioned avoidance via mental imagery. *Behaviour Research and Therapy*, *132*, 103652. https://doi.org/10.1016/j.brat.2020.103652

LaBar, K. S., & LeDoux, J. E. (1996). Partial disruption of fear conditioning in rats with unilateral amygdala damage: correspondence with unilateral temporal lobectomy in humans. *Behavioral Neuroscience*, *110*(5), 991–997. https://doi.org/10.1037//0735-7044.110.5.991

Lang, P. J. (1977). Imagery in therapy: an information processing analysis of fear. *Behavior Therapy*, *8*(5), 862–886. https://doi.org/10.1016/S0005-7894(77)80157-3

LeDoux, J. E., & Hofmann, S. G. (2018). The subjective experience of emotion: a fearful view. *Current Opinion in Behavioral Sciences*, *19*, 67–72. https://doi.org/10.1016/j.cobeha.2017.09.011

Liang, C., Li, Y., & Luo, J. (2016). A Novel Method to Detect Functional microRNA Regulatory Modules by Bicliques Merging. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *13*(3), 549–556. https://doi.org/10.1109/TCBB.2015.2462370

Mertens, G., Krypotos, A.-M., & Engelhard, I. M. (2020). A review on mental imagery in fear conditioning research 100 years since the “Little Albert” study. *Behaviour Research and Therapy*, *126*, 103556. https://doi.org/10.1016/j.brat.2020.103556

Morrison, S. E., & Salzman, C. D. (2010). Re-valuing the amygdala. *Current Opinion in Neurobiology*, *20*(2), 221–230. https://doi.org/10.1016/j.conb.2010.02.007

Mueller, E. M., Sperl, M. F. J., & Panitz, C. (2019). Aversive imagery causes de novo fear conditioning. *Psychological Science*, *30*(7), 1001–1015. https://doi.org/10.1177/0956797619842261

Muris, P., Bodden, D., Merckelbach, H., Ollendick, T. H., & King, N. (2003). Fear of the beast: a prospective study on the effects of negative information on childhood fear. *Behaviour Research and Therapy*, *41*(2), 195–208. https://doi.org/10.1016/s0005-7967(01)00137-1

Ono, T., Nishijo, H., & Uwano, T. (1995). Amygdala role in conditioned associative learning. *Progress in Neurobiology*, *46*(4), 401–422. https://doi.org/10.1016/0301-0082(95)00008-j

Pearson, J., & Kosslyn, S. M. (2015). The heterogeneity of mental representation: Ending the imagery debate. *Proceedings of the National Academy of Sciences of the United States of America*, *112*(33), 10089–10092. https://doi.org/10.1073/pnas.1504933112

Pearson, J., Naselaris, T., Holmes, E. A., & Kosslyn, S. M. (2015). Mental imagery: functional mechanisms and clinical applications. *Trends in Cognitive Sciences*, *19*(10), 590–602. https://doi.org/10.1016/j.tics.2015.08.003

Pfordresher, P. Q., Honda, C., Greenspon, E., & Chow, K. (2021). Generalization of novel sensorimotor associations among pianists and non-pianists: more evidence that musical training effects are constrained. *Psychological Research*, *85*(5), 1934–1942. https://doi.org/10.1007/s00426-020-01362-9

Reddan, M. C., Wager, T. D., & Schiller, D. (2018). Attenuating Neural Threat Expression with Imagination. *Neuron*, *100*(4), 994-1005.e4. https://doi.org/10.1016/j.neuron.2018.10.047

Schiller, D., Raio, C. M., & Phelps, E. A. (2012). Extinction training during the reconsolidation window prevents recovery of fear. *Journal of Visualized Experiments*, *66*, e3893. https://doi.org/10.3791/3893

Shin, L. M., & Liberzon, I. (2010). The neurocircuitry of fear, stress, and anxiety disorders. *Neuropsychopharmacology*, *35*(1), 169–191. https://doi.org/10.1038/npp.2009.83

Shi, Y., Zeng, W., Wang, N., Wang, S., & Huang, Z. (2015). Early warning for human mental sub-health based on fMRI data analysis: an example from a seafarers' resting-data study. *Frontiers in Psychology*, *6*, 1030. https://doi.org/10.3389/fpsyg.2015.01030

Singh, M. (2017). The cultural evolution of shamanism. *Behavioral and Brain Sciences*, 1–83. https://doi.org/10.1017/S0140525X17001893

Smolders, A., De Martino, F., Staeren, N., Scheunders, P., Sijbers, J., Goebel, R., & Formisano, E. (2007). Dissecting cognitive stages with time-resolved fMRI data: a comparison of fuzzy clustering and independent component analysis. *Magnetic Resonance Imaging*, *25*(6), 860–868. https://doi.org/10.1016/j.mri.2007.02.018

Soon, C.-S., Venkatraman, V., & Chee, M. W. L. (2003). Stimulus repetition and hemodynamic

response refractoriness in event-related fMRI. *Human Brain Mapping*, *20*(1), 1–12. https://doi.org/10.1002/hbm.10122

Supekar, K., Swigart, A. G., Tenison, C., Jolles, D. D., Rosenberg-Lee, M., Fuchs, L., & Menon, V. (2013). Neural predictors of individual differences in response to math tutoring in primary-grade school children. *Proceedings of the National Academy of Sciences of the United States of America*, *110*(20), 8230–8235. https://doi.org/10.1073/pnas.1222154110

Wang, J., Conder, J. A., Blitzer, D. N., & Shinkareva, S. V. (2010). Neural representation of abstract and concrete concepts: a meta-analysis of neuroimaging studies. *Human Brain Mapping*, *31*(10), 1459–1468. https://doi.org/10.1002/hbm.20950

Whelan, R., Watts, R., Orr, C. A., Althoff, R. R., Artiges, E., Banaschewski, T., Barker, G. J., Bokde, A. L. W., Büchel, C., Carvalho, F. M., Conrod, P. J., Flor, H., Fauth-Bühler, M., Frouin, V., Gallinat, J., Gan, G., Gowland, P., Heinz, A., Ittermann, B., … IMAGEN Consortium. (2014). Neuropsychosocial profiles of current and future adolescent alcohol misusers. *Nature*, *512*(7513), 185–189. https://doi.org/10.1038/nature13402

Yin, S., Liu, Y., Petro, N. M., Keil, A., & Ding, M. (2018). Amygdala Adaptation and Temporal Dynamics of the Salience Network in Conditioned Fear: A Single-Trial fMRI Study. *ENeuro*, *5*(1). https://doi.org/10.1523/ENEURO.0445-17.2018

Zhou, S., Cox, C. R., & Lu, H. (2019). Improving Whole-Brain Neural Decoding of fMRI with Domain Adaptation. In H.-I. Suk, M. Liu, P. Yan, & C. Lian (Eds.), *Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings* (Vol. 11861, pp. 265–273). Springer International Publishing. https://doi.org/10.1007/978-3-030-32692-0_31

Zhou, S., Li, W., Cox, C., & Lu, H. (2020). Side Information Dependence as a Regularizer for Analyzing Human Brain Conditions across Cognitive Experiments. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 6957–6964. https://doi.org/10.1609/aaai.v34i04.6179

Zhou, S. (2022). *Interpretable Domain-Aware Learning for Neuroimage Classification* [Doctoral dissertation]. University of Sheffield.

Vita

Lauryn Burleigh received a bachelor's degree from the University of New Orleans where their interest in cognitive research, particularly the use of fMRI, grew. Lauryn entered Louisiana State University under Dr. Steven Greening to further research in cognitive neuroscience using fRMI, obtaining their master's in Cognitive and Brain Science focusing on the neural mechanisms of fear to perceived and imagined stimuli. Through the program, Lauryn's interest in statistics and computational analyses continued. Lauryn began to focus on multivariate and machine learning analyses, furthering their understanding of data analysis. They completed their Ph.D. with Dr. Christopher Cox using a novel transfer learning function to further the analytical options of fMRI for fellow cognitive neuroscientists.