MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Improving Subject Transfer in EEG Classification with Divergence Estimation

Smedemark-Margulies, Niklas; Wang, Ye; Koike-Akino, Toshiaki; Liu, Jing; Parsons, Kieran; Bicer, Yunus; Erdogmus, Deniz

TR2025-044 April 02, 2025

Abstract

Classification models for electroencephalogram (EEG) data show a large decrease in performance when evaluated on unseen test subjects. We improve performance using new regularization techniques during model training. Approach. We propose several graphical models to describe an EEG classification task. From each model, we identify statistical relationships that should hold true in an idealized training scenario (with infinite data and a globally-optimal model) but that may not hold in practice. We design regularization penalties to enforce these relationships in two stages. First, we identify suitable proxy quantities (divergences such as Mutual Information and Wasserstein-1) that can be used to measure statistical independence and dependence relationships. Second, we provide algorithms to efficiently estimate these quantities during training using secondary neural network models. Main Results. We conduct extensive computational experiments using a large benchmark EEG dataset, comparing our proposed techniques with a baseline method that uses an adversarial classifier. We first show the performance of each method across a wide range of hyperparameters, demonstrating that each method can be easily tuned to yield significant benefits over an unregularized model. We show that, using ideal hyperparameters for all methods, our first technique gives significantly better performance than the baseline regularization technique. We also show that, across hyperparameters, our second technique gives significantly more stable performance than the baseline. The proposed methods require only a small computational cost at training time that is equivalent to the cost of the baseline. Significance. The high variability in signal distribution between subjects means that typical approaches to EEG signal modeling often require time-intensive calibration for each user, and even re-calibration before every use. By improving the performance of population models in the most stringent case of zero-shot subject transfer, we may help reduce or eliminate the need for model calibration.

Journal of Neural Engineering 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Improving Subject Transfer in EEG Classification with Divergence Estimation

Niklas Smedemark-Margulies[1]‡, Ye Wang[2], Toshiaki Koike-Akino[2], Jing Liu[2], Kieran Parsons[2], Yunus Bicer[3], Deniz Erdoğmuş[3]

[1]Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA
[2]Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA
[3]Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA

Abstract. *Objective.* Classification models for electroencephalogram (EEG) data show a large decrease in performance when evaluated on unseen test subjects. We improve performance using new regularization techniques during model training. *Approach.* We propose several graphical models to describe an EEG classification task. From each model, we identify statistical relationships that should hold true in an idealized training scenario (with infinite data and a globally-optimal model) but that may not hold in practice. We design regularization penalties to enforce these relationships in two stages. First, we identify suitable proxy quantities (divergences such as Mutual Information and Wasserstein-1) that can be used to measure statistical independence and dependence relationships. Second, we provide algorithms to efficiently estimate these quantities during training using secondary neural network models. *Main Results.* We conduct extensive computational experiments using a large benchmark EEG dataset, comparing our proposed techniques with a baseline method that uses an adversarial classifier. We first show the performance of each method across a wide range of hyperparameters, demonstrating that each method can be easily tuned to yield significant benefits over an unregularized model. We show that, using ideal hyperparameters for all methods, our first technique gives significantly better performance than the baseline regularization technique. We also show that, across hyperparameters, our second technique gives significantly more stable performance than the baseline. The proposed methods require only a small computational cost at training time that is equivalent to the cost of the baseline. *Significance.* The high variability in signal distribution between subjects means that typical approaches to EEG signal modeling often require time-intensive calibration for each user, and even re-calibration before every use. By improving the performance of population models in the most stringent case of zero-shot subject transfer, we may help reduce or eliminate the need for model calibration.

Keywords: Subject Transfer Learning, Brain-Computer Interface (BCI), Electroencephalography (EEG), Representation Learning, Domain Adaptation

Submitted to: *J. Neural Eng.*

‡ Work done while NSM and YB were interns at MERL

1. Introduction

In the field of signal modeling for electroencephalogram (EEG) and related biosignals, a key challenge is to train models that can extrapolate to unseen test subjects. It has been repeatedly observed in the literature [1] that signal models do not readily transfer to new subjects. Multiple factors contribute to this performance gap. First, non-invasive EEG measurements often include substantial noise and artifacts [2]. Second, supervised datasets for EEG and related biophysical measurements may contain noisy labels. In particular, such experiments often rely on subjects to adhere to visual timing cues, maintain attentiveness, or perceive and respond to a stimulus with a consistent timing; however human subjects may miss prompts, lose focus, or respond to incorrect stimuli due to confusion or perceptual errors. As a result, some data can be assigned the wrong task labels [3, 4]. Third, the distribution of brain signals can vary across subjects subjects (sometimes referred to as domain shift or covariate shift), such that pre-trained models may not be well-suited for unseen test subjects [5, 6, 7, 8].

High-level Approach. We introduce two new regularization methods to reduce this performance gap during subject transfer. Our methods are based on a pre-existing framework for subject transfer learning known as "censoring" [9]. While the benefits of the censoring framework have been demonstrated empirically in previous research, we provide new theoretical motivation, as well as new implementations that are simple and effective across a wide range of hyperparameters.

To derive a particular regularization penalty, we first select a generative model for the task and examine its conditional independence structure. We choose a statistical relationship that should hold true in an idealized classifier trained using data from this generative model, but which may not hold true in practice. We then convert this relationship to a regularization term by identifying a suitable quantity (a divergence such as mutual information or Wasserstein distance) to measure the relationship, and defining a simple algorithm for estimating this quantity during classifier training. By enforcing these relationships, censoring helps classifiers converge with less overfitting, despite being trained on a finite, noisy sample of data.

Experiments. We conduct extensive cross-validation experiments on a large benchmark EEG dataset to evaluate the effect of the proposed regularization methods.§ This benchmark dataset consists of binary EEG responses collected during a rapid serial visual presentation (RSVP) paradigm [10]. In each experiment, we train an EEG classifier model on a subset of subjects, with or without regularization, and measure the model's balanced accuracy on a set of unseen test subjects. To make a thorough statistical evaluation of our proposed methods, we perform over 60K such experiments, varying hyperparameters such as the regularization penalty, model structure, as well the set of training, validation, and test subjects, and the random initialization of the model.

The goal of these experiments is to measure how performance varies across hyperparameters, especially to compare the proposed methods against a baseline approach in terms of peak performance and stability of average performance. We measure test performance after a fixed training schedule, since this gives a direct comparison between a regularized and unregularized model. In supplementary materials, we also include experiments measuring test performance at the epoch of best validation accuracy; these secondary experiments evaluate how our techniques work in combination with early stopping regularization.

Our method includes several important hyperparameters; an explicit coefficient in our training loss, as well as several model design choices. Hyperparameters are typically tuned using a validation set [11]. This can be done using a variety of off-the-shelf software packages [12, 13, 14, 15]. Here, we present full results across a large hyperparameter grid and consider two key perspectives. First, we check whether there exist regions of hyperparameter space that give strong performance. If so, and if these regions are sufficiently large, then we may expect that practitioners could successfully apply our methods on future tasks after hyperparameter tuning. Second, we check whether performance is consistent across large regions of hyperparameter space. If so, then our methods may be useful even when careful hyperparameter tuning is not feasible (due to limited data or computational constraints).

§ Code to reproduce all experiments and analyses is available at: `https://github.com/merlresearch/eeg-subject-transfer`.

Results. When comparing the proposed methods to an unregularized model, we find that there are many regions of hyperparameters for which our methods provide significant improvements in balanced accuracy on unseen test subjects, and significantly improve generalization (measured as the ratio of test over train performance).

When comparing the proposed methods to a baseline censoring method, we find two advantages. First, we find that, given optimal hyperparameters for all methods, the density ratio censoring technique gives significantly higher balanced test accuracy than the baseline censoring method. This indicates that, with enough data and computational resources to perform hyperparameter tuning, the density ratio method is preferable to the baseline method. Second, the Wasserstein censoring technique leads to performance that is significantly more stable across hyperparameters than the baseline censoring method. This indicates that, when hyperparameter tuning is infeasible, the Wasserstein method may be preferable to the baseline. These benefits are most pronounced when measured after training for a fixed number of epochs, but supplementary experiments show these effects are still significant when training is also stopped early using validation metrics. This indicate that our method provides regularization that is partially separate from the effect of early stopping.

Contributions. The overall contributions of this work are as follows.

- We provide a novel theoretical motivation for a range of censoring regularization penalties.
- We derive two simple and efficient new estimation techniques for enforcing these regularization penalties, based on density ratio estimation and Wasserstein distances.
- Using extensive computational experiments, we find that one proposed technique significantly improves peak performance, and the other technique significantly increases stability of performance.

1.1. Related Work

Brain-computer interface research often focuses on restoring communication in individuals with severe speech and physical impairment (SSPI). Non-invasive electroencephalography (EEG) is a well-established modality for this purpose, with a wide variety of established experimental paradigms. A key limitation of these applications is the requirement for time-consuming calibration for new users or before each recording session. Our work focuses on reducing this time burden by improving zero-shot transfer performance.

In query-and-response paradigms, a subject is queried with a stimulus (such as images on a screen) and their EEG response is measured. In particular, we focus on a paradigm called rapid serial visual presentation (RSVP) [16]. Briefly, a subject first imagines a target item from a pre-defined set, such as one letter of the alphabet. The subject is queried with a sequence of multiple images in quick succession; each image in the sequence constitutes a binary trial, and contains one possible item from the pre-defined set. The subject's EEG response to each trial provides evidence about which symbol is desired. A symbol may be selected from one trial or query sequence, or the evidence from multiple sequences can be accumulated to perform recursive Bayesian inference [17]. For schematic explanations of the RSVP paradigm, see Figures 1 and 2 of Zhang et al. [10], or Figure 1 of Won et al. [18].

EEG is used for numerous other communication paradigms, including other query-and-response methods such as steady-state visually-evoked potentials (SSVEP) [19], and paradigms without a stimulus prompt such as motor imagery (MI) [20] or classification of emotional affect [21]. Subject transfer learning is a common challenge across these communication paradigms and for the modeling of related biosignals data types such as electromyography (EMG) and electrocorticography (ECoG) [22].

Some work on subject transfer learning has applied domain adaptation methods, with the goal of harmonizing datasets from different subjects, measurement devices, or experimental paradigms. The goal in these approaches is to be able to train a single model on these collected datasets [23, 24, 25].

Previous work has investigated the use of censoring penalties in training variational autoencoders [26] and learning disentangled representations[27]. Other work has applied censoring penalties to enforce different notions of conditional independence, using estimation techniques such as kernel density estimation and neural critic functions [28]. Our work extends these approaches by providing a theoretical motivation for each censoring penalty and providing two new methods for estimating censoring penalties that are highly effective and simple to implement.

We compare our proposed methods to a widely-used baseline technique that uses a secondary adversarial classifier model to guide regularization. We select this baseline because it has been applied in the same censoring framework that we use to derive our new methods [9], and is therefore directly comparable. Furthermore, this baseline is conceptually simple, easy to implement, and has shown widespread success and adoption. Specifically, variations on this method have been used to harmonize feature

distributions in a range of transfer learning studies [29, 30, 31]. Many variations of this technique have been successfully applied for representation learning and transfer learning in EEG classification [9, 26, 32, 33, 34, 35, 36].

The algorithms we develop here rely on several techniques from the generative modeling literature. One technique uses density ratio estimation [37] to approximately compute a Mutual Information (MI) term; a similar technique has been previously demonstrated for other applications [38]. More recent work has explored other approaches to estimating MI [39]. Our other technique replaces the use of Kullback-Leibler (KL) divergence with Wasserstein-1 distance in order to estimate dependence between variables. This has been previously described as a Wasserstein dependency measure [40]; our approach to computing an estimate of the Wasserstein-1 distance is based on previous research on sampling realistic images [41].

2. Methods

Overview. Here, we define the unseen subject classification task and motivate our approach. We provide three generative models describing this task. From these generative models, we select one or more statistical relationships at a time to enforce during model training for regularization; we refer to each choice of one relationship as a "censoring mode." We formally define the components of our model architecture. Next, we introduce several estimation techniques for measuring the statistical relationships that we hope to enforce, and show how to train our model with the desired regularization. Finally, we describe the computational experiments that we perform to evaluate our proposed methods.

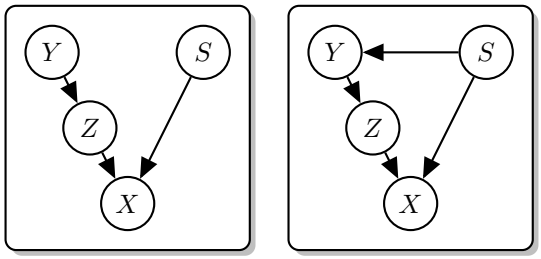
2.1. Problem Statement and Motivation

Consider a dataset of tuples $\{(x, y, s)\}$, with data $x \in \mathbb{R}^D$, discrete task labels $y \in \{1, \ldots, C\}$, and discrete nuisance labels $s \in \{1, \ldots, S\}$. The nuisance labels represent the combination of subject identifier and session identifier. These tuples will be sampled from an empirical data distribution $(x, y, s) \sim p(X, Y, S)$, whose generative model is described below. We seek to train a classifier on a subset of subjects, and regularize the model's training to achieve high accuracy on unseen test subjects. At test time, we will receive only a set of data X from the test subject, and must infer the corresponding set of task labels Y.

Idealized and Real-world Settings. In order to train a classifier to infer $p(Y|X)$, we can first choose a generative model describing how we believe the dataset was produced. If this generative model matches the true generating process for the dataset, and if training results in a classifier that is well-fit to the dataset, then we would expect to find that the trained classifier exhibits the same statistical relationships that exist in the generative model. For example, we would expect that variables which are independent in the generative model are also independent in the distribution learned by the classifier. In an idealized setting where we have infinite, unbiased training data and a global optimization algorithm, we may expect this favorable outcome (where the learned model matches the generative model) with no additional effort.

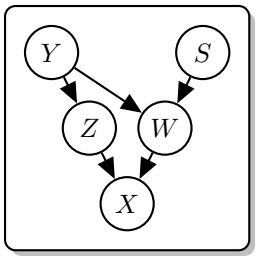
In practice, however, we typically encounter several key limitations. Tasks involving biosignals such as EEG often have very limited training data that is both noisy and may not reveal representative features. Furthermore, typical classifiers are trained using local optimization strategies, such as using stochastic gradient descent on a non-convex loss function. Thus, we do not expect models trained using only a classification objective to necessarily obey the correct dependence structure. In particular, note that models for biosignals classification tasks may incorrectly learn a distribution of features that correlates strongly with the subject identifier [32]; essentially a form of overfitting to the training set. This may explain the common experimental observation of a "subject transfer gap" - a large decrease in model performance when tested on unseen subjects [42].

We reduce this subject transfer gap using regularization penalties. By specifying a certain generative model, we have also implicitly defined a set of statistical relationships such as conditional independences. We can easily enumerate these relationships, e.g. using the "Bayes Ball" algorithm [43]. For a pair of variables A and B, conditioned on a set of zero or more additional observed variables C, we may identify that our model implies relationships such as a marginal independence $A \perp B$, a conditional independence $A \perp B|C$, or a conditional dependence $A \not\perp B|C$. Note that the set of all such statements is combinatorially large in the number of individual variables of the generative model; thus it is not feasible to enforce them all. We select just one or two of these statistical relationships at a time, and enforce them as a regularization objective. This approach helps the model converge to a better optimum that will generalize to unseen subjects with less overfitting. We refer to these regularization objectives as "censoring" objectives, because we deliberately choose relationships involving the model's latent features and the nuisance labels.



(a) $Z \perp S$

(b) $Z \perp S|Y$



(c) $Z \perp S$, $W \not\perp S$

Figure 1: Graphical models for EEG classification that motivate different regularization approaches. (a): the distribution of actions does not differ across subjects $p(Y|S) = p(Y)$; introducing a latent variable Z facilitates regularization by enforcing *marginal* independence $Z \perp S$. (b): actions may vary across subjects $p(Y|S) \neq p(Y)$; this correlation suggests enforcing *conditional* independence $Z \perp S|Y$. (c): a second latent variable is introduced to capture nuisance-related information for use inferring task labels; *complementary* regularization is performed with a pair of penalties to enforce independence $Z \perp S$ and dependence $W \not\perp S$.

2.2. Graphical Models and Censoring modes.

Figure 1 shows three possible graphical models for an EEG classification task, each of which motivates a different regularization strategy.

In Figure 1a, we consider the case of a single latent variable Z and define the generative process as

$$p(X, Y, Z) = p(S)p(Y)p(Z|Y)p(X|S, Z). \quad (1)$$

By introducing a latent variable, we separate the step of extracting useful features from the step of predicting task labels, and define a space in which we can perform regularization. In this first model, the latent variable should be marginally independent of the nuisance labels $Z \perp S$, giving the first censoring mode which we refer to as **marginal censoring**. This model makes the simplifying assumption that the distribution of task labels does not differ across different subjects or sessions, so that there is no direct link between S and Y (i.e. $p(Y|S) = p(Y)$).

Figure 1b relaxes this assumption and adds a connection from S to Y; the resulting generative process is defined as

$$p(X, Y, Z) = p(S)p(Y|S)p(Z|Y)p(X|S, Z). \quad (2)$$

This dependence could arise in an EEG typing task where a subject tends to use their preferred letters or words with higher frequency. The connection between S and Y means that the latent variable is no longer marginally independent of the nuisance variable; we instead enforce conditional independence $Z \perp S|Y$, giving our second censoring mode called **conditional censoring**. Intuitively, this allows the latent features to have some information about the nuisance variable, but no more than the amount already implied by Y.

In Figure 1c, to address the possibility that the nuisance variable may be informative when predicting the task label at test time, we include a second latent variable W that captures nuisance-related information. The generative process becomes

$$p(X, Y, Z) = p(S)p(Y)p(Z|Y)p(W|Y, S)p(X|Z, W). \quad (3)$$

Recall that for unseen test subjects, the value of S will not be available. Furthermore, its value would not be directly useful to the model, since it comes from a region of the domain of S that was never observed during training. Instead, we hope to infer the second latent variable W, including some nuisance-related information, from the data X; this may help the classifier model to better predict Y. In this model, one latent variable is marginally independent of the nuisance variable $Z \perp S$, while the other is strongly determined by the nuisance variable, which we merely describe as $W \not\perp S$ (note that we try to maximize this dependence in our penalties, even though this notation requires only a minimal correlation). This censoring mode is called **complementary censoring**.

2.3. Model Architecture

To approach the unseen subject classification task, we construct a task model to classify data and a censoring model to regularize the task model, as shown in Figure 2.

For convenience, let $p(Z, Y, S)$ denote the distribution obtained by sampling from the empirical data distribution $(x, y, s) \sim p(X, Y, S)$ and then applying the encoder and projector $z = P_{\theta_P}(F_{\theta_F}(x))$. Define $p(Z, S)$ as the same pushforward distribution after marginalizing over Y (easily achieved by dropping y after sampling); likewise define $p(Z)$ by marginalizing over both Y and S.

The task model consists of an encoder $F_{\theta_F}(\cdot) : \mathbb{R}^D \to \mathbb{R}^K$ that produces K-dimensional *hidden* features $\tilde{z} \in \mathbb{R}^K$, and a classifier $G_{\theta_G}(\cdot) : \mathbb{R}^K \to \Delta(C)$ that maps feature vectors to a vector in

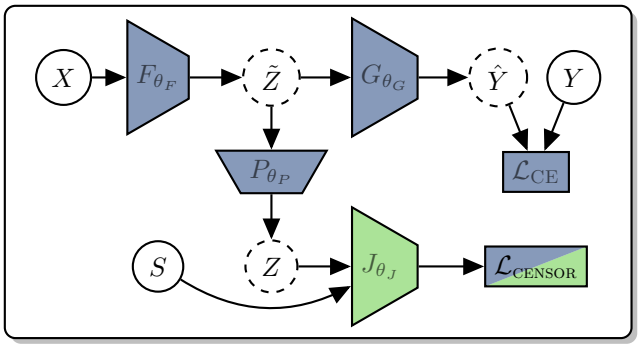


Figure 2: Model Architecture. Trapezoids are trainable models: encoder F_{θ_F}, classifier G_{θ_G}, projection P_{θ_P}, and censoring model J_{θ_J}. Solid circles are input variables: data X, true task labels Y, and nuisance labels S. Dashed circles are intermediate variables: hidden features $\tilde{Z}$, observed features Z, and predicted task labels $\hat{Y}$. Rectangles are loss terms: cross-entropy loss $\mathcal{L}_{\text{CE}}$ and regularization penalty $\mathcal{L}_{\text{CENSOR}}$. Training alternates between updating blue and green model components; both $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{CENSOR}}$ are used to update the main model, while the censoring model is only trained using $\mathcal{L}_{\text{CENSOR}}$ (with appropriate changes such as inverted sign; see below). J_{θ_J} receives additional inputs in some settings.

the C-dimensional probability simplex. The task model is trained so that predicted label distribution $\hat{y} = G_{\theta_G}(F_{\theta_F}(x)), x \sim p(X)$ approximates the true posterior distribution over labels $y \sim p(Y|X)$. The projection $P_{\theta_P}(\cdot) : \mathbb{R}^K \to \mathbb{R}^K$ maps *hidden* feature vectors to *observed* feature vectors $\tilde{z} \mapsto z$; this is included based on empirical benefits observed in the contrastive learning literature [44]. In some experiments, this projection is the identity mapping with zero parameters, which we refer to as "trivial" or "direct features"; otherwise, the projection is "non-trivial" and gives "projected features." During training, the projection is updated along with the task model.

The censoring model $J_{\theta_J}(\cdot) : \mathbb{R}^T \to \mathbb{R}^L$ regularizes the task model. The censoring model's input and output dimensions vary between censoring modes and estimation algorithms. Its input may include half ($T = K/2$) or all of the latent features ($T = K$), or it may include one-hot encoded nuisance values ($T = K + |S|$) or task labels ($T = K + |S| + C$). Its output may be a scalar value ($L = 1$), or a predicted probability vector over nuisance labels ($L = |S|$). Sections 2.6, 2.7, or 2.8 explain the structure of this model in more detail as well as how it is applied to the projected features to compute a regularization penalty. The censoring model's parameters are updated in an alternating optimization against the task model.

2.4. Unregularized and Regularized Training

In the empirical risk minimization (ERM) framework, the *risk* $R(\theta)$ is defined as the expected loss of a model for a particular set of parameters θ, when applied to samples from a particular dataset and evaluated using a chosen loss function [45]. For convenience, let $\mathcal{Q}_Y \coloneqq q_{\theta_F,\theta_G}(Y|X)$ represent the label posterior estimated by our task model $G_{\theta_G}(F_{\theta_F}(X))$, and let $\mathcal{P}_Y \coloneqq p(Y|X)$ represent the empirical label posterior. We use the cross-entropy loss $\mathcal{L}_{\text{CE}}$, and thus we can define the empirical risk as:

$$R(\theta_F, \theta_G) = \mathop{\mathbb{E}}_{p(X,Y,S)} \Big[\mathcal{L}_{\text{CE}}(\mathcal{P}_Y, \mathcal{Q}_Y) \Big]. \tag{4}$$

To find optimal parameters, we minimize the risk: $\min_{\theta_F,\theta_G} R(\theta_F, \theta_G)$.

2.5. Deriving Censoring objectives

In the censoring framework, we add a regularization term $\mathcal{L}_{\text{CENSOR}}$ to the optimization problem above. Note this regularization term depends on the parameters θ_F and θ_P of the task model's encoder and projector, and the parameters θ_J of the censoring model. To find the optimal regularized parameters, we minimize the regularized risk:

$$\min_{\theta_F,\theta_G,\theta_P} R(\theta_F, \theta_G) + \lambda \max_{\theta_J} \mathcal{L}_{\text{CENSOR}}. \tag{5}$$

The purpose of this regularization term is to help enforce one or more statistical relationships that we expect should hold true, according to the generative model we assume for the task. To obtain a tractable penalty, these statistical relationships (such as $Z \perp S$) must be converted into concrete quantities that we can estimate or compute analytically, such as a mutual information or a divergence between two distributions. Then, we can create algorithms to estimate these concrete quantities. Finally, we can use these estimates in our regularization objective while training our model.

Implications of Introducing Hyperparameters Note that using a regularized objective introduces several hyperparameters. There is an explicit hyperparameter λ in the modified objective, and discrete model design choices implicit in the auxiliary loss term $\mathcal{L}_{\text{CENSOR}}$. Since the overall performance of the model will be sensitive to these choices, applying our methods to new tasks will likely require hyperparameter tuning. Standard techniques for hyperparameter optimization are well reviewed elsewhere [11]), but the conceptually simplest approach is to withhold a validation dataset, perform a grid search in which hyperparameters are varied one at a time, and select the hyperparameter

values that yield best performance. While it can be challenging and resource-intensive to perform tuning in this way, many techniques exist for adaptively adjusting search ranges and step sizes on each parameter, and these techniques are available in off-the-shelf software toolkits [12, 13, 15, 14]

Here, we perform a large hyperparameter grid search in order to broadly characterize our proposed methods. We consider two potential successful cases for a method with important hyperparameters. On one hand, if there exist regions of hyperparameter space that achieve strong performance, then the method will be useful, provided tuning is feasible. On the other hand, if tuning is not feasible, then a method that offers consistent benefit across hyperparameters may be more useful.

Using divergences to measure statistical relationships. The three censoring modes that we consider each reflect a particular statement about the dependence or independence of variables. We consider two concrete quantities that can be used to measure dependence between variables; mutual information (MI) and Wasserstein-1 (W_1) distance. In general, we can replace a statement about the independence of two variables A and B with a statement about the statistical divergence between the joint distribution $p(A, B)$ and the product of marginal distributions $p(A)p(B)$. This comparison is often made using the Kullback-Leibler (KL) divergence, which yields Mutual Information (MI) $I(A; B)$. We consider several ways to estimate MI in order to enforce independence (and dependence) relationships in Section 2.6 and 2.7. However, this comparison may also be made using other measures; in Section 2.8, we replace KL divergence with the Wasserstein-1 (W_1) metric.

2.6. Adversarial Classifier Baseline

As a baseline regularization method, we consider a well-studied approach where the censoring model $J_{\theta_J}(\cdot)$ is an adversarial classifier (see Section 1.1 for more background).

Marginal Censoring Algorithm 1 describes how to compute the regularization penalty in (5) using this adversarial classifier method for the case of marginal censoring. Recall that, in this case, we seek to enforce $Z \perp S$; to achieve this, we will compute a regularization penalty $\mathcal{L}_{\text{CENSOR}}$ that approximates $I(Z; S)$.

The adversarial classifier is trained alongside the task model in an alternating optimization scheme; its objective is to use the observed latent features Z to predict the nuisance label S. Intuitively, if the MI between these variables is high, the adversary will be able to predict the nuisance label well, and thus the adversary's classification performance can serve as a proxy measure for the mutual information $I(Z; S)$.

Algorithm 1: Marginal Censoring using Adversarial Classifier

Input: Tuples of data, label, nuisance $\{(x_i, y_i, s_i)\}_{i=1}^N$, encoder F_{θ_F}, projector P_{θ_P}, adversarial classifier J_{θ_J}

Output: $\mathcal{L}_{\text{CENSOR}}$ approximating $I(Z; S)$

for $i \in 1 \dots N$ **do**
- $\tilde{z}_i \leftarrow F_{\theta_F}(x_i)$ `// Encode`
- $z_i \leftarrow P_{\theta_P}(\tilde{z}_i)$ `// Project`
- $\mathcal{L}_i \leftarrow \mathcal{L}_{\text{CE}}\big(q_{\theta_J}(s_i|z_i),\ p(s_i|z_i)\big)$

return – $avg(\mathcal{L})$ `// Mean CE loss`

For convenience, here let $\mathcal{Q}_S \coloneqq q_{\theta_J}(S|Z)$ refer to the censoring model's predicted distribution over nuisance labels, and let $\mathcal{P}_S \coloneqq p(S|Z)$ refer to the corresponding ground-truth (one-hot) distribution. We can see that the censoring model's cross-entropy loss $\mathcal{L}_{\text{CE}}\big(\mathcal{P}_S,\ \mathcal{Q}_S\big)$ serves as a lower bound on $I(Z; S)$, as follows. Note that the MI can be decomposed as $I(Z; S) = H(S) - H(S|Z)$. The marginal entropy $H(S)$ is constant during our optimization process, since it only depends on the data distribution. We can obtain a bound on the other term, the conditional entropy $H(S|Z)$, by writing out the definition of cross entropy:

$$\mathcal{L}_{\text{CE}}(\mathcal{P}_S, \mathcal{Q}_S) = H(S|Z) + \underbrace{\text{KL}\big(\mathcal{P}_S \parallel \mathcal{Q}_S\big)}_{\geq 0} \geq H(S|Z). \quad (6)$$

Thus we can relate the censoring model's cross-entropy and the MI we seek to minimize:

$$I(Z; S) = H(S) - H(S|Z) \geq H(S) - \mathcal{L}_{\text{CE}}(\mathcal{P}_S, \mathcal{Q}_S). \quad (7)$$

In order to enforce $Z \perp S$, we seek to minimize $I(Z; S)$; however if we minimize $\mathcal{L}_{\text{CE}}(\mathcal{P}_S, \mathcal{Q}_S)$ as a proxy, we are actually minimizing a lower bound on the desired quantity. As shown above, this bound will be close when $\text{KL}\big(\mathcal{P}_S \parallel \mathcal{Q}_S\big)$ is small, which may occur when the censoring model is sufficiently flexible and trained to convergence.

Training a regularized model using the adversarial classifier involves alternating between updating the parameters of the censoring model using

$$\theta_J^* = \arg\min_{\theta_J} \mathcal{L}_{\text{CE}}(\mathcal{P}_S, \mathcal{Q}_S), \quad (8)$$

and updating the parameters of the task model using (5) where the regularization penalty $\mathcal{L}_{\text{CENSOR}}$ is obtained using Algorithm 1.

Conditional and Complementary Censoring This method can also be used for conditional censoring. Recall that in conditional censoring we seek to enforce $Z \perp S|Y$. This corresponds to reducing the conditional MI $I(Z;S|Y) = H(S|Y) - H(S|Z,Y)$. We can modify the censoring model so that it takes both features and task label as input, and tries to predict the conditional probability over nuisance labels; let $q_{\theta_J}(S|Z,Y)$ represent the output of this modified censoring model. The first term $H(S|Y)$ is constant with respect to our optimization process; as before, the second term $H(S|Z,Y)$ can be bounded by the cross entropy $\mathcal{L}_{\text{CE}}\big(p(S|Z,Y),\ q_{\theta_J}(S|Z,Y)\big)$ using an analogous derivation. Thus the censoring model's cross-entropy again gives us a bound on the desired MI term.

In the case of complementary censoring, recall that we seek to enforce one independence relationship $Z \perp S$ and one dependence relationship $W \not\perp S$; we achieve this by applying the same censoring model twice. For the first set of latent features Z, we use the same procedure as in the marginal censoring case; for the second set of latent features W, we use the same procedure and invert the sign of the final regularization term. This results in an objective of the form

$$\min_{\theta_F,\theta_G,\theta_P} R(\theta_F,\theta_G) + \lambda \max_{\theta_J}(\mathcal{L}_{\text{CENSOR,Z}} - \mathcal{L}_{\text{CENSOR,W}}), \quad (9)$$

where $\mathcal{L}_{\text{CENSOR,Z}}$ regularizes Z and $\mathcal{L}_{\text{CENSOR,W}}$ regularizes W.

2.7. Density Ratio Censoring

As described above, in the adversarial classifier approach, the adversary's cross-entropy loss provides a lower bound on one or more mutual information terms. Here, the censoring model $J_{\theta_J}(\cdot)$ is trained to directly estimate the mutual information.

Density Ratio Estimation We first briefly introduce a method for density ratio estimation established in the generative modelling literature [37]. Given two distributions over the same space $p(x)$ and $q(x)$, we can estimate the log ratio of their densities $\log\big(p(x)/q(x)\big)$ by training a binary classifier C to distinguish between samples from p versus q. By minimizing the cross-entropy objective,

$$\min_C \mathop{\mathbb{E}}_{p(x)}\big[-\log\sigma(C(x))\big] + \mathop{\mathbb{E}}_{q(x)}\big[-\log\sigma(-C(x))\big], \quad (10)$$

where $\sigma(z) = 1/(1+e^{-z})$, and $C(x)$ is the logit of the binary classifier, we obtain an optimal classifier C^* whose output is the desired log ratio $C^*(x) = \log\frac{p(x)}{q(x)}$. In the case of generating synthetic data, the objective in (10) is used to train a discriminator between samples of the true data distribution and the synthetic data distribution [46, 47, 48, 49].

Algorithm 2: Computing Training Loss for Density Ratio Estimator

Input: Tuples of data, label, nuisance $\{(x_i, y_i, s_i)\}_{i=1}^N$, encoder F_{θ_F}, projector P_{θ_P}, density ratio estimator J_{θ_J}

Output: Loss for training θ_J

$\tilde{S} \leftarrow permute(S)$
for $i \in 1 \dots N$ **do**
  $\tilde{z}_i \leftarrow F_{\theta_F}(x_i)$ `// Encode`
  $z_i \leftarrow P_{\theta_P}(\tilde{z}_i)$ `// Project`
  $\mathcal{L}_i^{\text{JOINT}} \leftarrow -\log\sigma(J_{\theta_J}(z_i, s_i))$ `//` $p(Z,S)$
  $\mathcal{L}_i^{\text{PROD}} \leftarrow -\log\sigma(-J_{\theta_J}(z_i, \tilde{s}_i))$ `//` $p(Z)p(S)$
return $avg(\mathcal{L}_{\text{JOINT}}) + avg(\mathcal{L}_{\text{PROD}})$ `// Eq` (12)

Marginal Censoring This density ratio estimation technique can be directly applied for estimating the mutual information between two variables; the censor model J_{θ_J} plays the role of the binary classifier C above. Algorithm 2 describes how to train this density ratio estimator model. Recall that mutual information is defined as an expected log-likelihood ratio

$$I(Z;S) := \mathop{\mathbb{E}}_{p(Z,S)}\left[\log\frac{p(Z,S)}{p(Z)p(S)}\right]. \quad (11)$$

The censoring model's training objective is,

$$\min_{\theta_J} \mathop{\mathbb{E}}_{p(Z,S)}[-\log\sigma(J_{\theta_J}(Z,S))] + \mathop{\mathbb{E}}_{p(Z)p(S)}[-\log\sigma(-J_{\theta_J}(Z,S))], \quad (12)$$

such that J_{θ_J} learns to approximate $\log\frac{p(Z,S)}{p(Z)p(S)}$. Note that this training objective requires samples from the empirical joint distribution $p(Z,S)$ as well as from the product of marginal distributions $p(Z)p(S)$. Samples from $p(Z)p(S)$ can be approximated by simply permuting one of the variables. To see that this shuffling gives the desired samples, consider first sampling and encoding a batch of items $\{Z,Y,S\}$ and discarding Y,S. This gives an approximate sample from the marginal distribution $p(Z)$, whose order is unimportant. Likewise sample items from $p(S)$ by discarding Z,Y and optionally shuffling. By sampling one batch and only shuffling S, we perform these two processes in one step.

The density ratio estimator model can then be used to approximate mutual information as

$$I(Z;S) \approx \mathop{\mathbb{E}}_{p(Z,S)}[J_{\theta_J}(Z,S)]. \quad (13)$$

The overall procedure for training with density ratio censoring involves alternating between updating the parameters θ_J of the censoring using Algorithm 2, and

Algorithm 3: Marginal Censoring using Density Ratio Estimator

Input: Tuples of data, label, nuisance $\{(x_i, y_i, s_i)\}_{i=1}^N$, encoder F_{θ_F}, projector P_{θ_P}, density ratio estimator J_{θ_J}

Output: $\mathcal{L}_{\text{CENSOR}}$ approximating $I(Z;S)$

for $i \in 1 \dots N$ **do**
- $\tilde{z}_i \leftarrow F_{\theta_F}(x_i)$ `// Encode`
- $z_i \leftarrow P_{\theta_P}(\tilde{z}_i)$ `// Project`
- $\mathcal{L}_i \leftarrow J_{\theta_J}(z_i, s_i)$ `// Eq` (13)

return $avg(\mathcal{L})$

updating the parameters of the task model using (5), where the regularization penalty $\mathcal{L}_{\text{CENSOR}}$ is given by Algorithm 3.

Conditional and Complementary Censoring To perform conditional censoring using the density ratio estimation method, we adjust the training objective for θ_J from (12) as follows. We seek to enforce the conditional independence $Z \perp S|Y$, which corresponds to minimizing the conditional mutual information $I(Z;S|Y)$. By chain rule of mutual information, we have $I(Z;S|Y) = I(Z,Y;S) - I(Y;S)$. Since $I(Y;S)$ is fixed with respect to our optimization process, $I(Z,Y;S)$ is a suitable proxy to minimize. In order to estimate this quantity, we first adjust the censoring model to accept three inputs instead of two. The definition of MI states that

$$I(Z,Y;S) := \mathop{\mathbb{E}}_{p(Z,Y,S)} \left[\log \frac{p(Z,Y,S)}{p(Z,Y)p(S)} \right]. \tag{14}$$

We can estimate the inner log density ratio $\log \frac{p(Z,Y,S)}{p(Z,Y)p(S)}$ by training the censor model with

$$\min_{\theta_J} \mathop{\mathbb{E}}_{p(Z,Y,S)} [-\log \sigma(J_{\theta_J}(Z,Y,S))] + \mathop{\mathbb{E}}_{p(Z,Y)p(S)} [-\log \sigma(-J_{\theta_J}(Z,Y,S))]. \tag{15}$$

This objective requires samples from $p(Z,Y)p(S)$, which we can obtain by shuffling the nuisance labels within a batch (analogous to the shuffling trick for the marginal case).

To perform complementary censoring, we use the marginal censoring approach twice; once to estimate $I(Z;S)$, and a second time to estimate $I(W;S)$. The resulting objective has the same form as the complementary censoring objective in (9).

2.8. Wasserstein Censoring

In the previous two sections, we enforce independence (or dependence) by minimizing (maximizing) an estimate of mutual information. Here, we replace mutual information with the Wasserstein-1 (W_1) distance between a joint distribution and a product of marginal distributions.

For two variables A and B, the chain rule of probability states that the joint distribution can always be expressed as $p(A,B) = p(A)p(B|A)$. If A and B are independent, then $p(B|A) = p(B)$, and the joint distribution $p(A,B)$ equals the product of marginals $p(A)p(B)$. Whereas mutual information measures the distance between $p(A,B)$ and $p(A)p(B)$ using the KL divergence, any other notion of statistical divergence may be used to similar effect. Following previous work in the generative modeling literature, we consider the Wasserstein-1 (W_1) metric; this approach has been previously described as a Wasserstein dependency measure [40].

Marginal Censoring. To apply this for marginal censoring, we seek to measure the W_1 distance between $p(Z,S)$ and $p(Z)p(S)$. Under the Kantorovich-Rubinstein duality theorem [50], this distance is

$$\mathrm{W}_1(r,q) = \sup_{\|f\|_L \le 1} \mathop{\mathbb{E}}_{r}[f(Z,S)] - \mathop{\mathbb{E}}_{q}[f(Z,S)], \tag{16}$$
$$\text{where } r := p(Z,S) \text{ and } q := p(Z)p(S).$$

Note that the "critic" function f has Lipschitz norm bounded by 1. As established in the generative modeling literature, the critic function f can be implemented using be a neural network with an arbitrary Lipschitz constant K, giving an estimate of $K\mathrm{W}_1(\cdot,\cdot)$ that suffices in practice for minimizing or maximizing $\mathrm{W}_1(\cdot,\cdot)$[41]. We satisfy this requirement in the standard fashion using spectral normalization [51] on each layer of the critic network. Note that (16) requires samples from $p(Z)p(S)$; we use the same trick as in the Section 2.7 of shuffling the nuisance variable within a batch to obtain such samples. Algorithm 4 describes how we can use a critic neural network to estimate the Wasserstein distance in (16) in order to perform marginal censoring. Note that the critic model receives two inputs. Training a model using Wasserstein censoring involves alternating between updates to the parameters of the critic model θ_J and the parameters of the task model; when updating the task model, the output from Algorithm 4 is used directly; when updating the critic model, the same loss is used with the sign flipped.

Conditional and Complementary Censoring To perform conditional censoring using the Wasserstein method, we adjust the training scheme described above as follows. First, the critic model is adjusted to accept three inputs (observed features Z, task labels Y, and nuisance labels S). Next, we begin with the same logic

Algorithm 4: Marginal Censoring using Wasserstein Critic

Input: Tuples of data, label, nuisance $\{(x_i, y_i, s_i)\}_{i=1}^N$, encoder F_{θ_F}, projector P_{θ_P}, Wasserstein critic J_{θ_J}

Output: $\mathcal{L}_{\text{CENSOR}}$ approximating $\mathrm{W}_1\big(p(Z,S),\ p(Z)p(S)\big)$

$\tilde{S} \leftarrow permute(S)$

for $i \in 1 \dots N$ **do**

  $\tilde{z}_i \leftarrow F_{\theta_F}(x_i)$ `// Encode`

  $z_i \leftarrow P_{\theta_P}(\tilde{z}_i)$ `// Project`

  $\mathcal{L}_i^{\text{JOINT}} \leftarrow J_{\theta_J}(z_i, s_i)$ `//` $p(Z,S)$

  $\mathcal{L}_i^{\text{PROD}} \leftarrow J_{\theta_J}(z_i, \tilde{s}_i)$ `//` $p(Z)p(S)$

return $avg(\mathcal{L}^{\text{JOINT}}) - avg(\mathcal{L}^{\text{PROD}})$

as in the case of conditional censoring using the density ratio estimator method (see Section 2.7). For that method, we showed that the conditional independence $Z \perp S|Y$ can be enforced by minimizing $I(Z;S|Y)$, and in turn this can be replaced by minimizing $I(Z,Y;S)$. We used this final quantity because we can easily obtain samples from the relevant distributions ($p(Z,Y,S)$ and $p(Z,Y)p(S)$). Here, we replace the use of KL divergence in $I(Z,Y;S)$ with $\mathrm{W}_1\big(p(Z,Y,S),\ p(Z,Y)p(S)\big)$, which we estimate using a critic neural network as in Algorithm 4 and Equation (4).

In complementary censoring, the Wasserstein critic is used twice, once to minimize $\mathrm{W}_1(p(Z,S), p(Z)p(S)$, and once to maximize $\mathrm{W}_1(p(W,S), p(W)p(S))$.

2.9. Computational Experiments

Dataset We use a large publicly-available EEG dataset for all experiments [10]. This dataset contains EEG recordings during a rapid serial visual presentation (RSVP) task with binary trials. Subjects were presented with a sequence of quickly flashed images and asked to watch for target images, while their EEG responses were recorded. Each stimulus presentation is associated with a binary label. Data were recorded at 1000Hz and made available at a downsampled rate of 250Hz. The dataset includes just over 1 million binary trials, collected from 64 subjects, each of whom participated in 2 recording sessions.

Experimental Setup In each experiment, we evaluated the performance of a single proposed regularized training method, defined by the parameters listed in Table 1. Models were trained for a fixed number of epochs using all sessions of data from 28 subjects for training, and using all sessions of data from 4 subjects for testing.

We used cross-validation to obtain reliable estimates of model performance. Each experiment was repeated 100 times using 10 different initial random seeds and 10 different choices of train/val/test subject assignment. Note that the dataset contains 64 total subjects, while each experiment used 32 subjects; thus the 10 subject splits are partially overlapping. Model performance was quantified using balanced accuracy, which is the average of accuracy on each class.

In addition to the data X and binary task labels Y, experiments require a nuisance label S, computed as an integer that uniquely identifies a particular subject and session. Non-target trials were subsampled to achieve a proportion of 10 non-target trials per 1 target trial to be similar to real-world RSVP applications such as assistive typing.

Hyperparameters Explored Table 1 summarizes the hyperparameters varied across experiments.

Hyperparameter	**Range Explored**
Censor Mode	Marginal, Conditional, Complementary
Censor Method	Adversarial Classifier, Density Ratio Estimator, Wasserstein Critic
Censor Strength (λ)	0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.5, 1, 2, 3, 5, 10, 20, 30, 50, 100.0
Projection Type	Trivial ($P_{\theta_P} = I$), Non-trivial
Evaluation Point	Final Checkpoint, Best Val Checkpoint

Table 1: Hyperparameters varied across experiments. Each experiment was repeated 100 times, using 10 random seeds and 10 splits of train, validation, and test subjects. *Censor Mode*: choice of graphical model and statistical relationship to enforce (see Section 2.2). *Censor Method*: technique used to compute regularization penalty (see Sections 2.6, 2.7, and 2.8). *Projection Type*: whether projection network P_{θ_P} is the identity function (see Figure 2. *Censor Strength*: value of coefficient λ in (5). *Evaluation Point*: whether model is evaluated at epoch of best validation accuracy, or final (100th) epoch.

For marginal and conditional censoring, the dimension of $\tilde{Z}$ and Z was 128. For complementary censoring, 64 dimension were used for Z and 64 for W. Models were implemented and trained using PyTorch [52] and Pytorch Lightning [53], using the AdamW optimizer [54] with constant learning rate 10^{-4}, default values of $\beta_1 = 0.9, \beta_2 = 0.999$, and batch size 1024, for 100 epochs.

The encoder was a 1D convolutional network with 248K parameters θ_F. The classifier was a multi-layer

perceptron (MLP) with 50K parameters θ_G. When present, the projection network was an MLP with 66K parameters θ_P. The censoring model was an MLP, with between 48K and 56K parameters θ_J, depending on the number of input vectors (Z, S for marginal censoring, Z, Y, S for conditional, Z, Y and W, Y with $dim(Z) = dim(W) = 64$ for complementary) and the dimension of the output ($dim(S)$ for the adversarial classifier method; 1D for the density ratio estimation and Wasserstein censoring methods).

Cross-Validation To obtain a stable estimate of the effect of our proposed methods, each experiment was run 100 times, using 10 cross-validation folds for each of 10 random seeds. This helped control for variation due to the particular assignment of subjects into train, validation, and test sets, as well as variation due to weight initialization and batch selection during training. Note that in a single cross-validation fold, the subjects used for train, validation, and test are all disjoint.

Statistical Evaluation After collecting the results of our experiments, we perform several forms of statistical evaluation.

First, for each choice of hyperparameters, we obtain a collection of results that we compare to the unregularized model using two-sided paired t-tests, since each individual result corresponds to using the same random seed and same data split for the the regularized and unregularized models. This allows us to determine whether each method provides an advantage over the unregularized model at each setting. We did not perform any correction for multiple hypothesis testing; since tests performed are not independent, optimal correction is non-trivial. In particular, the performance of one censoring method using a certain hyperparameter value λ_1 may be highly related to the performance using a nearby value $\lambda_1 + \epsilon$.

Second, we compare the peak performance of each regularization method. We find the optimal value of λ in each censoring mode and pool the results into three groups (one representing each censoring method). Since the optimal values of λ are not necessarily the same, a paired may not be suitable; furthermore, it may not be reasonable to assume equal variance between groups. Therefore, we compare these pooled values using two-sided Welch's t-tests [55]. This allows us to compare aggregate performance of our two proposed methods against the baseline method, in the scenario that careful hyperparameter tuning has been applied to each.

Third, we compare the variability of mean performance across hyperparameters, by taking the mean performance value for every value of λ across all censoring modes, and pooling these into three groups (one representing each method). We compare these values using Levene's test for unequal variance [56]. This allows us to determine whether the average performance of our proposed methods is more stable across hyperparameters than the baseline, in the scenario that careful hyperparameter tuning would not be feasible.

3. Results

To highlight the severity of the subject transfer gap in this dataset, note that the unregularized model achieves a mean balanced accuracy on train data of 99%, but a mean balanced accuracy on test data of only 68%.

3.1. Balanced Accuracy across Hyperparameters

Figure 3 shows the distribution of balanced accuracy on the test set at the end of training. The top panel shows the baseline method, the middle panel shows the proposed density ratio method, and the bottom panel shows the proposed Wasserstein method. In each panel, a group of boxplots on the X-axis represents a single choice of censoring mode and projection type (e.g. marginal censoring with a trivial projection). Each single boxplot represents a single value of λ in (5), and shows 100 repetitions of the experiment across different data folds and random seeds. The unregularized model's performance is shown by horizontal black lines; solid lines show lower quartile, median, and upper quartile, while the dashed line shows the mean. A two-sided paired t-test was performed between the 100 balanced accuracy scores of each censored model and the 100 scores of the unregularized model, and annotations are added when the censored model's mean is larger. No symbol is added when the unregularized model's mean is larger, though in some cases this difference is also significant.

To provide context for our hyperparameter tuning experiments, we first describe in a simplified manner the space of possible outcomes that could occur and highlight some possible outcomes of interest when evaluating the results of hyperparameter tuning.

Recall that we have a training objective with two terms (Eq. (5)). At very low values of λ, the regularization loss term is nearly multiplied by 0 and will have no effect; thus the regularized model should perform almost the same as the unregularized model. At intermediate values of λ, if the regularization is beneficial, we may see a peak in performance where the regularized model outperforms the unregularized model; we refer to such ideal values as λ^*; if the regularization is not helpful, then there will be no such peak in performance. Finally, for very large values of

λ, the task loss term will no longer have a strong effect on training, and the model's performance may decrease due to the excessive regularization.

In general, when examining the results of such a hyperparameter sweep, we are interested to see whether a peak in performance occurs for some λ^* (indicating that a regularization technique can be useful), whether this occurs for a narrow region of λ (indicating the need for very careful hyperparameter tuning to achieve a benefit) or a broad region of λ (indicating that hyperparameter tuning may be performed relatively easily), and finally whether excessive regularization (for $\lambda > \lambda^*$) is harmful to model performance (indicating whether the technique could be safely applied even without hyperparameter tuning).

For all three censoring methods, we observe that very small λ values give performance that closely matches the unregularized model. This indicates the low end of our λ search is sufficiently small to capture the full range of behavior. Since this low end of λ is almost never below the unregularized model, when searching for ideal λ^* value, it is generally safe to use any $\lambda \leq \lambda^*$. Across the three methods, we observe a peak where performance is improved for most censor modes, though the ideal value λ^* varies between methods and censor modes, indicating that hyperparameter searching would be required to use any of these methods on a new dataset. The trend at large λ differs between the methods.

For the density ratio method, the performance at optimal values λ^* is particularly high, and consistent across censoring modes. In each peak region, there are several contiguous λ values that yield strong performance benefits; thus we may expect that hyperparameter searching for a new dataset may be successful (even when using a log-linear scale over a very large range, as was done here). Importantly, when λ is too large (meaning that very strong regularization is applied), the density ratio method can also greatly decrease performance.

For the Wasserstein method, we again see that there is a region of peak performance in each censoring mode for some value λ^*, and in this region the method offers a significant benefit compared to an unregularized model. Interestingly, in most cases, we do not observe any negative consequences from using values $\lambda > \lambda^*$.

For the baseline adversarial classifier method, we see a strong benefit for the ideal λ value in the conditional and complementary censoring modes, but the peak is reduced or absent in the marginal censoring mode.

3.2. Comparing Peak Performance across Methods

In Figure 4, we compare the aggregate performance of each method using optimal hyperparameter tuning. For each method, we pool values from the best value of λ for each censor mode and projection type. This includes 600 values for each censor method. As in Figure 3, the unregularized model's 100 performance values are shown by horizontal black lines. We compare each of our proposed methods against the adversarial censoring baseline using the two-sided Welch's t-test. We find that peak performance for the density ratio method is significantly greater than the baseline ($p = 0.0354$). We find that peak performance for the Wasserstein method is slightly lower than the baseline, though the difference is not significant ($p = 0.0624$).

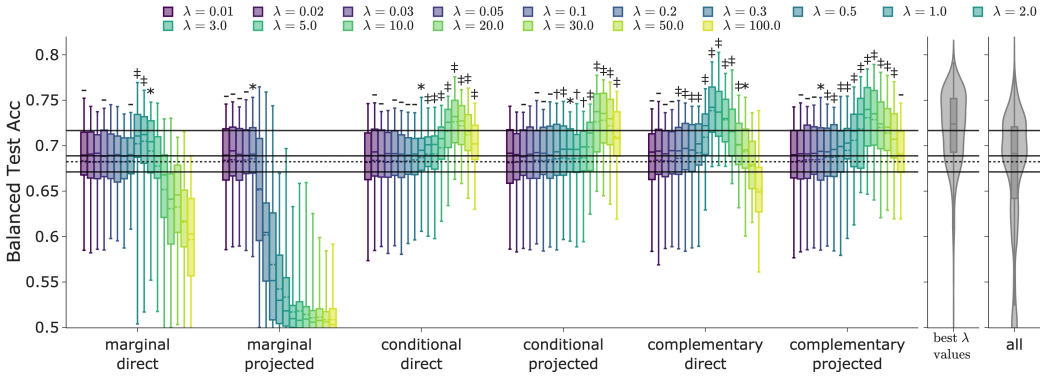
3.3. Comparing Variability across Methods

In Figure 5, we group the mean performance from each censor mode and all values of λ, for each method. This allows us to compare the variability in mean performance across hyperparameters. To emphasize the difference from Figures 3 and 4, we show the individual points (each representing the mean performance for a certain λ, censor mode, and projection type), and do not display boxplots. Furthermore, we have only a single corresponding point for the unregularized model's performance (the mean of its 100 runs), shown as a horizontal black line. To compare variability across groups, we use Levene's test of unequal variance. While the variance of the density ratio method's mean performance (0.00162) is about three times lower than the variance of the adversarial method's mean (0.00302), this difference is not significant ($p = 0.847$). By contrast, we observe that the variance of the Wasserstein method's mean (0.0002) is significantly lower than the adversarial method ($p < 0.001$) This observation indicates that the Wasserstein method's performance is more stable across hyperparameter choices, and may indicate that making large changes to hyperparameters is safer when using this method.

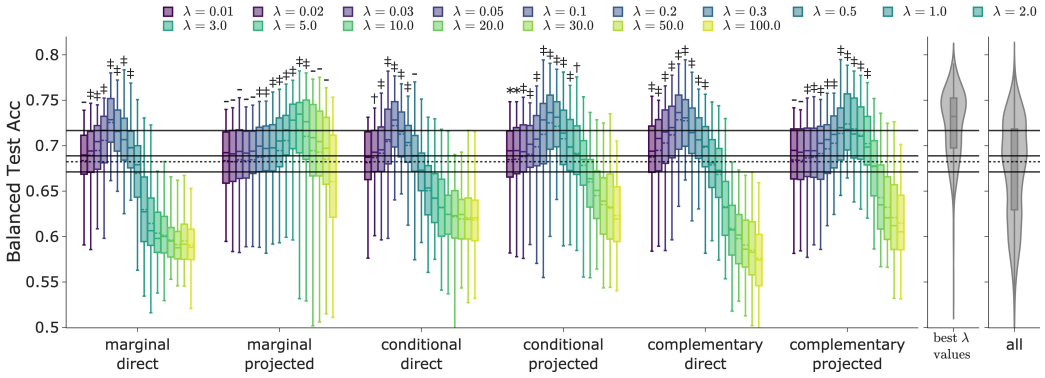
3.4. Comparing Generalization Ratios across Methods

To measure the effect of regularization on overfitting, we compute a "generalization ratio," which we define to be the balanced accuracy on test data divided by the balanced accuracy on train data. This ratio describes the performance decrement that can be expected from train to test set. A ratio of 1 would indicate that test performance matched train performance and the model did not overfit; a ratio less than 1 would indicate that the model was overfit to the training set.

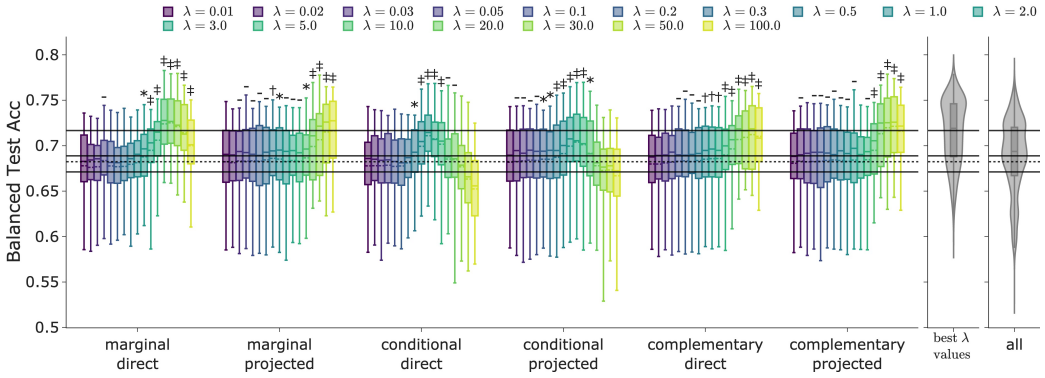
In Figure 6, we select the best value of λ from each censor mode and projection type (the same



(a) Adversarial Censoring (Baseline)



(b) Density Ratio Censoring



(c) Wasserstein Censoring

Figure 3: Hyperparameter sweep of balanced test accuracy. (a): adversarial classifier baseline (Sec 2.6). (b): density ratio censoring (Sec 2.7). (c): Wasserstein censoring (Sec 2.8). Boxplots show 100 trials, varying random seed and data split. Horizontal black lines show unregularized model performance. When regularized model's mean exceeds unregularized model's, a symbol annotation shows significance from a two-sided paired t-test (-, $p > 0.05$; *, $0.01 < p \leq 0.05$; †, $0.001 < p \leq 0.01$; ‡, $p \leq 0.001$). Left violin plot ('best λ values') shows pooled results from best λ of each censor mode and projection type. Right violin plot ('all') shows all data pooled. *Marginal*, *conditional*, *complementary*: censoring modes (Sec 2.2). *Projected*: projection model P_{θ_P} is non-trivial; *direct*: P_{θ_P} is omitted. λ: strength of regularization in (5).

collection of experiments used in Figure 4) and plot the distribution of generalization ratios. The unregularized model's generalization ratios are shown by horizontal black lines.

We use the two-sided Welch's t-test to compare performance between groups. In these experiments, we find that neither method is significantly different from the baseline (density ratio $p = 0.393$; Wasserstein

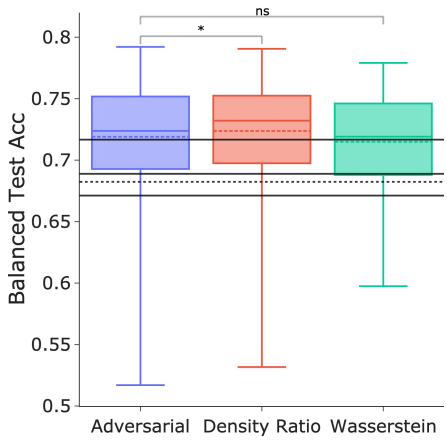


Figure 4: Peak performance across methods. Each method includes results from the λ values with best mean balanced accuracy in each censor mode and projection type (same values used in left violin plot of Fig. 3). Horizontal black lines show unregularized model performance. Annotations show results of Welch's t-test (ns, non-significant; *, $p \leq 0.05$).

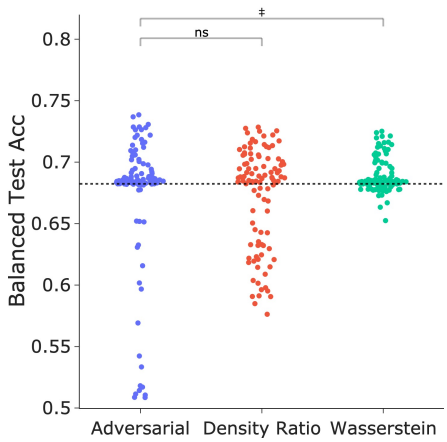


Figure 5: Stability of performance across methods. Each point is mean of a single boxplot in Fig. 3 (one censor mode, projection type, and λ value). Horizontal black line represents mean of unregularized model. Annotations show significance of Levene's test for unequal variance (ns, non-significant; ‡, $p \leq 0.001$).

$p = 0.571$). Since we saw in Figure 4 that the balanced test accuracies for the density ratio method were significantly higher than for the adversarial method, this indicates that the density ratio method's training accuracies were also higher. Both the baseline and the newly proposed censoring methods achieve generalization ratios dramatically higher than the unregularized model; this shows that all three methods are able to greatly reduce overfitting.

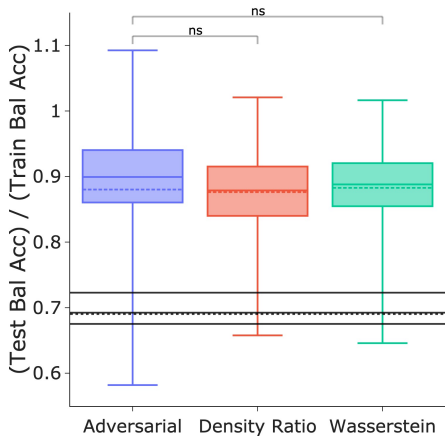


Figure 6: Generalization ratio across methods. Each method includes results from the λ values with best mean balanced accuracy in each censor mode and projection type (same values used in left violin plot of Fig. 3 and in Fig. 4). Horizontal black lines show unregularized model performance. Annotations show results of two-sided Welch's t-test (ns, non-significant; *, $p \leq 0.05$).

4. Discussion

We study the problem of regularized model training to perform zero-shot subject transfer learning for EEG classification tasks. We provide a novel motivation for the censoring regularization strategy. Two assumptions must be met for classifier models to achieve high performance: the dataset being used for training must match the assumed generative model for the task, and the classifier model must learn the dependency structure implied by this generative model. When we observe low model performance, this might occur because one or both of these assumptions is violated. We provide regularization penalties to address the second source of error. Specifically, for any particular generative model, we select a statistical relationship that should hold, convert this to a divergence that should be minimized (here, a mutual information term or a Wasserstein distance), and then add this as a regularization term in the training objective. By considering several graphical models and their conditional independence structure, we identify three different statistical relationships that can be enforced to regularize the model.

In computational experiments on a large benchmark EEG dataset, we found that our techniques significantly improve performance compared to an unregularized model across a wide range of hyperparameters. When using optimal hyperparameters, we found that the benefits of the proposed density ratio method were significantly greater than the benefits of a baseline

adversarial classifier method. Across the full range of hyperparameters explored, we found that mean performance of the proposed Wasserstein method was significantly more stable than the adversarial baseline. While all three methods significantly reduce overfitting compared to an unregularized model, we did not find significant differences in overfitting between methods.

5. Conclusion

Significance In this work, we provide a theoretically well-founded, end-to-end procedure to obtain a useful regularization penalty for an EEG classification task. The benefits of the proposed methods for unseen subject transfer learning may help reduce the burden of calibration time in BCI applications.

Future Work The current techniques may be adapted to use other quantitative measures of statistical dependence. When comparing the joint distribution and product of marginals as a means of estimating dependence between two variables, we considered KL divergence (leading to the mutual information measure) and Wasserstein-1 distance (leading to the Wasserstein critic technique). Any other measure of statistical distance or divergence would also be suitable, such as other f-divergences [57, 58, 59], Maximum Mean Discrepancy [60], or other methods for estimating Wasserstein distance and related measures such as Sinkhorn divergences [61]. Depending on the experimental context and availability of calibration data for an unseen subject, models trained using our method may be fine-tuned using standard techniques, as reviewed in previous literature [62].

As mentioned previously, there are a variety of standard techniques and software tools for offline hyperparameter tuning using validation data. Future work may consider techniques for online and adaptive tuning of hyperparameters, to accommodate gradual changes in data characteristics or user behavior.

Our work may be extended to new tasks by adjusting the graphical model and following the same procedure to derive a regularization algorithm.

Limitations The proposed regularization strategies have several important drawbacks. To obtain the largest benefit, these techniques should be used with careful hyperparameter tuning. Such tuning requires allocating a held-out validation set, and can be computationally expensive, depending on the size of models, datasets, and the range of hyperparameters search. The proposed Wasserstein method offers stable performance across hyperparameters; however, using these or other regularization methods with inappropriate hyperparameters can cause a decrease in task performance. Using these regularization penalties also incurs a small computational cost during training, when compared to the unregularized model, since there is an alternating optimization with a small secondary model. While the present study made thorough examination of hyperparameter sensitivity, only one dataset and one underlying model architecture were explored. We focus on measuring the relative performance of different models; absolute performance could be further improved by tuning standard hyperparameters such as model architecture and data preprocessing steps.

References

[1] Dongrui Wu, Yifan Xu, and Bao-Liang Lu. "Transfer learning for EEG-based brain–computer interfaces: A review of progress made since 2016". In: *IEEE Transactions on Cognitive and Developmental Systems* 14.1 (2020), pp. 4–19.

[2] Chi Qin Lai et al. "Artifacts and noise removal for electroencephalogram (EEG): A literature review". In: *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*. IEEE. 2018, pp. 326–332.

[3] Erin Gibson et al. "EEG variability: Task-driven or subject-driven signal of interest?" In: *NeuroImage* 252 (2022), p. 119034.

[4] Anne K Porbadnigk et al. "When brain and behavior disagree: Tackling systematic label noise in eeg data with machine learning". In: *2014 International Winter Workshop on Brain-Computer Interface (BCI)*. IEEE. 2014, pp. 1–4.

[5] Simanto Saha and Mathias Baumert. "Intra- and inter-subject variability in EEG-based sensorimotor brain computer interface: a review". In: *Frontiers in computational neuroscience* 13 (2020), p. 87.

[6] Jakub Šťastný, Pavel Sovka, and Milan Kostilek. "Overcoming Inter-Subject Variability In BCI Using EEG-Based Identification." In: *Radioengineering* 23.1 (2014).

[7] Chun-Shu Wei et al. "A subject-transfer framework for obviating inter-and intra-subject variability in EEG-based drowsiness detection". In: *NeuroImage* 174 (2018), pp. 407–419.

[8] Bo-Qun Ma et al. "Reducing the subject variability of EEG signals with adversarial domain generalization". In: *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part I 26*. Springer. 2019, pp. 30–42.

[9] Ye Wang, Toshiaki Koike-Akino, and Deniz Erdogmus. "Invariant representations from adversarially censored autoencoders". In: *arXiv preprint arXiv:1805.08097* (2018).

[10] Shangen Zhang et al. "A benchmark dataset for RSVP-based brain–computer interfaces". In: *Frontiers in neuroscience* 14 (2020), p. 568000.

[11] Tong Yu and Hong Zhu. "Hyper-parameter optimization: A review of algorithms and applications". In: *arXiv preprint arXiv:2003.05689* (2020).

[12] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[13] Takuya Akiba et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2623–2631.

[14] James Bergstra, Dan Yamins, David D Cox, et al. "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms". In: *Proceedings of the 12th Python in science conference*. Vol. 13. Citeseer. 2013, p. 20.

[15] Richard Liaw et al. "Tune: A research platform for distributed model selection and training". In: *arXiv preprint arXiv:1807.05118* (2018).

[16] Stephanie Lees et al. "A review of rapid serial visual presentation-based brain–computer interfaces". In: *Journal of neural engineering* 15.2 (2018), p. 021001.

[17] Niklas Smedemark-Margulies et al. "Recursive Estimation of User Intent From Noninvasive Electroencephalography Using Discriminative Models". In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2023, pp. 1–5.

[18] Kyungho Won et al. "EEG dataset for RSVP and P300 speller brain-computer interfaces". In: *Scientific Data* 9.1 (2022), p. 388.

[19] Anthony M Norcia et al. "The steady-state visual evoked potential in vision research: A review". In: *Journal of vision* 15.6 (2015), pp. 4–4.

[20] Piotr Wierzgała et al. "Most popular signal processing methods in motor-imagery BCI: a review and meta-analysis". In: *Frontiers in neuroinformatics* 12 (2018), p. 78.

[21] Edgar P Torres et al. "EEG-based BCI emotion recognition: A survey". In: *Sensors* 20.18 (2020), p. 5083.

[22] Vinay Jayaram et al. "Transfer learning in brain-computer interfaces". In: *IEEE Computational Intelligence Magazine* 11.1 (2016), pp. 20–31.

[23] Marco Congedo, Alexandre Barachant, and Rajendra Bhatia. "Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review". In: *Brain-Computer Interfaces* 4.3 (2017), pp. 155–174.

[24] Bingchuan Liu et al. "Align and pool for EEG headset domain adaptation (ALPHA) to facilitate dry electrode based SSVEP-BCI". In: *IEEE Transactions on Biomedical Engineering* 69.2 (2021), pp. 795–806.

[25] Wei-Long Zheng and Bao-Liang Lu. "Personalizing EEG-based affective models with transfer learning". In: *Proceedings of the twenty-fifth international joint conference on artificial intelligence*. 2016, pp. 2732–2738.

[26] Ozan Özdenizci et al. "Transfer learning in brain-computer interfaces with adversarial variational autoencoders". In: *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE. 2019, pp. 207–210.

[27] Mo Han et al. "Disentangled adversarial transfer learning for physiological biosignals". In: *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2020, pp. 422–425.

[28] Niklas Smedemark-Margulies et al. "AutoTransfer: Subject transfer learning with censored representations on biosignals data". In: *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2022, pp. 3159–3165.

[29] Yaroslav Ganin et al. "Domain-adversarial training of neural networks". In: *The journal of machine learning research* 17.1 (2016), pp. 2096–2030.

[30] Eric Tzeng et al. "Adversarial discriminative domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7167–7176.

[31] Mingsheng Long et al. "Conditional adversarial domain adaptation". In: *Advances in neural information processing systems* 31 (2018).

[32] Ozan Özdenizci et al. "Adversarial deep learning in EEG biometrics". In: *IEEE signal processing letters* 26.5 (2019), pp. 710–714.

[33] Bo-Qun Ma et al. "Depersonalized cross-subject vigilance estimation with adversarial domain generalization". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–8.

[34] Samaneh Nasiri and Gari D Clifford. "Attentive adversarial network for large-scale sleep staging". In: *Machine Learning for Healthcare Conference*. PMLR. 2020, pp. 457–478.

[35] Xingliang Tang and Xianrui Zhang. "Conditional adversarial domain adaptation neural network for motor imagery EEG decoding". In: *Entropy* 22.1 (2020), p. 96.

[36] He Zhao et al. "Deep representation-based domain adaptation for nonstationary EEG classification". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.2 (2020), pp. 535–545.

[37] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. "Density ratio estimation: A comprehensive review (statistical experiment and its related topics)". In: 1703 (2010), pp. 10–31.

[38] Taiji Suzuki et al. "Approximating mutual information by maximum likelihood density ratio estimation". In: *New challenges for feature selection in data mining and knowledge discovery*. PMLR. 2008, pp. 5–20.

[39] Ben Poole et al. "On variational bounds of mutual information". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5171–5180.

[40] Sherjil Ozair et al. "Wasserstein dependency measure for representation learning". In: *Advances in Neural Information Processing Systems* 32 (2019).

[41] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[42] Zitong Wan et al. "A review on transfer learning in EEG signal analysis". In: *Neurocomputing* 421 (2021), pp. 1–14.

[43] Ross D Shachter. "Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams)". In: *arXiv preprint arXiv:1301.7412* (2013).

[44] Kartik Gupta et al. "Understanding and Improving the Role of Projection Head in Self-Supervised Learning". In: *arXiv preprint arXiv:2212.11491* (2022).

[45] Vladimir Vapnik. "Principles of risk minimization for learning theory". In: *Advances in neural information processing systems* 4 (1991).

[46] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. "Estimating divergence functionals and the likelihood ratio by convex risk minimization". In: *IEEE Transactions on Information Theory* 56.11 (2010), pp. 5847–5861.

[47] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. "f-gan: Training generative neural samplers using variational divergence minimization". In: *Advances in neural information processing systems* 29 (2016).

[48] Yuchen Pu et al. "Adversarial symmetric variational autoencoder". In: *Advances in neural information processing systems* 30 (2017).

[49] Benjamin Rhodes, Kai Xu, and Michael U Gutmann. "Telescoping density-ratio estimation". In: *Advances in neural information processing systems* 33 (2020), pp. 4905–4916.

[50] Cédric Villani et al. *Optimal transport: old and new*. Vol. 338. Springer, 2009.

[51] Takeru Miyato et al. "Spectral normalization for generative adversarial networks". In: *arXiv preprint arXiv:1802.05957* (2018).

[52] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[53] William Falcon and The PyTorch Lightning team. *PyTorch Lightning*. Version 1.8.6. Mar. 2019. DOI: 10.5281/zenodo.3828935. URL: https://github.com/Lightning-AI/lightning.

[54] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[55] Bernard L Welch. "The generalization of 'STUDENT'S' problem when several different population varlances are involved". In: *Biometrika* 34.1-2 (1947), pp. 28–35.

[56] Howard Levene. "Robust tests for equality of variances". In: *Contributions to probability and statistics* (1960), pp. 278–292.

[57] Alfréd Rényi. "On measures of entropy and information". In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. Vol. 4. University of California Press. 1961, pp. 547–562.

[58] Paul Rubenstein et al. "Practical and consistent estimation of f-divergences". In: *Advances in Neural Information Processing Systems* 32 (2019).

[59] Sreejith Sreekumar and Ziv Goldfeld. "Neural estimation of statistical divergences". In: *The Journal of Machine Learning Research* 23.1 (2022), pp. 5460–5534.

[60] Arthur Gretton et al. "A kernel two-sample test". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 723–773.

[61] Aude Genevay, Gabriel Peyré, and Marco Cuturi. "Learning generative models with sinkhorn divergences". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 1608–1617.

[62] Wonjun Ko et al. "A survey on deep learning-based short/zero-calibration approaches for EEG-based brain–computer interfaces". In: *Frontiers in Human Neuroscience* 15 (2021), p. 643386.

6. Appendix

6.1. Effect of Censoring on Generalization Ratio

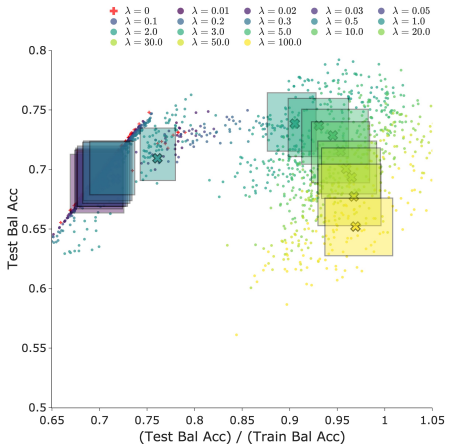
As described in Figure 6, we quantify the effect of regularization on model generalization by computing a "generalization ratio," which we define as the ratio of balanced accuracy on test data divided by balanced accuracy on train data.

Figure 7 shows plots of balanced test accuracy on the vertical axis, and generalization ratio on the horizontal axis. Due to space constraints, results of only a few selected model hyperparameters are shown. Each plot shows the performance of a single censoring method for one choice of mode and projection type; this corresponds to one X-axis group from Figure 3. Just as with Figure 3, the colors indicate changing values of λ. Here, the unregularized model (with $\lambda = 0$) is shown in red. Note that the Y-axis coordinates here show the same values shown in Figure 3, while the X-axis shows the generalization ratio. For each value of λ, colored points show the 100 independent reruns across data folds and random seeds; the colored box represents the interquartile range (IQR) of the points along each axis. The unregularized model ($\lambda = 0$) is also shown in each plot.

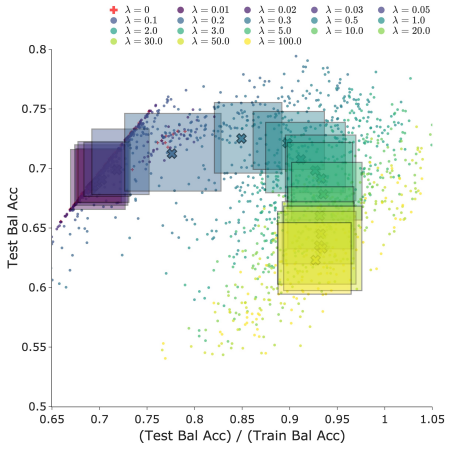
An ideal model would have a large Y-axis value, indicating strong test performance, and a large X-axis value, indicating that it retains its training performance when transferring to unseen test subjects. As λ increases, we observe that the distribution for all three censoring methods moves towards the upper-right, indicating improvements in both metrics. Beyond an ideal value of λ, the distributions change direction and move towards the lower-right, indicating that regularization is too strong. Specifically; excess regularization brings both training and test accuracy down at about the same rate. In general, a method which can be easily tuned would show gradual movement on this plot; we observe that the adversarial baseline has larger jumps in X-axis coordinate as λ changes, though we have not quantified this effect.

6.2. Experiments Combining Censoring with Early Stopping

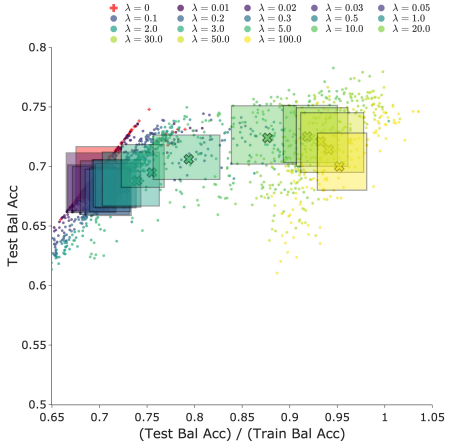
In our main experiments, models were trained for a fixed number of epochs, and we compared performance of each regularization method to the unregularized model, and to each other. Here, we also present results showing the performance of models when early stopping is also performed using a held-out validation set. Early stopping is a commonly used regularization technique that is both easy to apply and very effective, though it requires sacrificing a portion of training data, and therefore may sometimes be infeasible. The



(a) Adversarial Censoring (Baseline)



(b) Density Ratio Censoring



(c) Wasserstein Censoring

Figure 7: Test Balanced Accuracy vs Generalization Ratio. Ideal models have large X- and Y-coordinate. (a): adversarial censoring, complementary mode, without P_{θ_P}. (b): density ratio censoring, conditional mode, using P_{θ_P}. (c): Wasserstein censoring, marginal mode, without P_{θ_P}. Points show 100 trials, varying random seed and data split. Boxes show interquartile range on each axis.

purpose of these experiments is to explore whether the regularization benefits of each method censoring are subsumed by the regularization benefits of early stopping. These experiments may be informative when considering whether it is useful to try censoring regularization in conjunction with early stopping (or other regularization techniques). Note that early stopping based on validation performance requires sacrificing a portion of training data, and may not be applicable in some settings.

Whereas in the main experiments models were trained on data from 28 subjects and tested on data from 4 subjects, in these early-stopping experiments, models were trained with all sessions of data from 24 subjects for training, 4 subjects were withheld for validation, and 4 subjects were used for testing. The model checkpoint from the epoch of best validation performance was used for testing. Training lasted up to 30 epochs (since the point of optimal early stopping almost always occurs before this).

6.2.1. Balanced Accuracy across Hyperparameters Figure 8 shows results analogous to Figure 3, but models were tested using the checkpoint of best validation performance. This optimal early stopping already provides a strong regularization to both unregularized and censored models, reducing the potential incremental benefit of censoring.

The most apparent trend here is that the benefit of censoring has been greatly reduced, because the "unregularized" model is now actually regularized by early stopping, and achieves much higher balanced accuracy on test data.

As before, we are also interested in the shape of performance as the hyperparameter λ varies. We again see that, for small enough values, performance is mostly unaffected; and that for excess values of λ, task performance suffers due to over-regularization. In this case, we find that the density ratio method still shows significant benefits in all censor modes and projection types, compared to the model with only early stopping. For the adversarial censoring baseline, there are a few choices of censor mode, projection type, and λ that give significant benefit over the early-stopped model. For Wasserstein censoring, there are many cases that give a non-significant benefit, but none that give significant benefit. Another difference between these results and the main results is that the point where λ becomes too large has been shifted left; this indicates that, in the presence of another form of regularization, it is important to err on the side of smaller λ when tuning hyperparameters.

We make quantitative comparison of the peaks and variability of these trends below, but these results generally shows that the density ratio method has the highest peaks, the Wasserstein method has the least variability across λ.

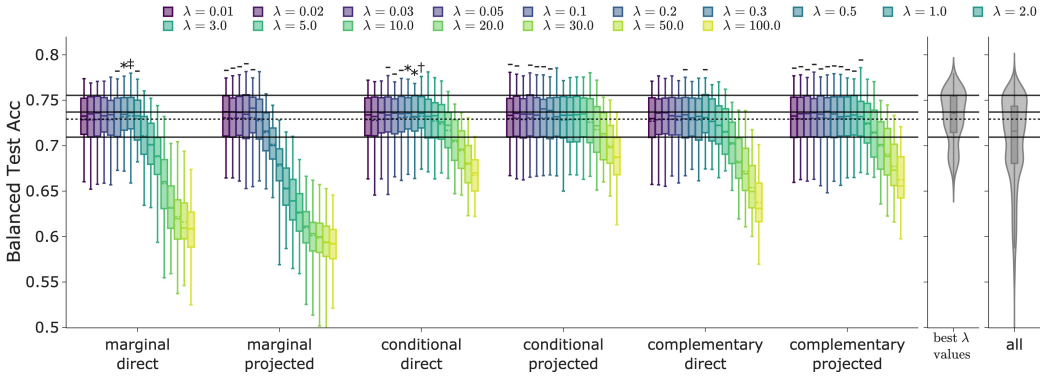
6.2.2. Comparing Peak Performance across Methods Figure 9 shows results analogous to Figure 4, but when censoring was combined with optimal early stopping as described above. From each censor mode and projection type, we selected the λ value giving the best mean test performance. These pooled results were compared using the two-sided Welch's t-test. Here, we find the same trend as observed in our main experiments. The density ratio method gives significantly higher peak performance than the adversarial baseline ($p = 0.030$), while the Wasserstein method's peak performance was not significantly different from the adversarial baseline ($p = 0.411$).

6.2.3. Comparing Variability across Methods Figure 10 shows results analogous to Figure 5, but using both censoring and optimal early stopping. We group the mean performance from each censor mode and all values of λ for each method to measure variability in the mean performance across hyperparameters.

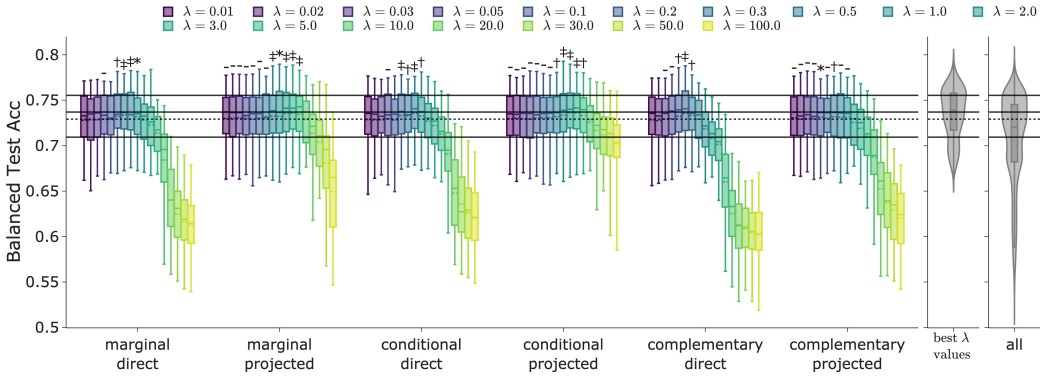
We observe the same trends here as in Figure 5. The variability of mean performance for the adversarial baseline method and density ratio method are not significantly different ($p = 0.971$), but variability is significantly lower for the Wasserstein method than the baseline ($p = 1e-5$). This demonstrates again that performance of the Wasserstein method is more stable across hyperparameters than performance of the other two censoring methods tested.

6.2.4. Comparing Generalization Ratios across Methods Figure 11 shows results analogous to Figure 6, when both censoring and optimal early stopping are applied. We select the best value of λ from each censor mode and projection type (the same collection of experiments used in Figure 9) and plot the distribution of generalization ratio. The model with early stopping only is shown by horizontal black lines.

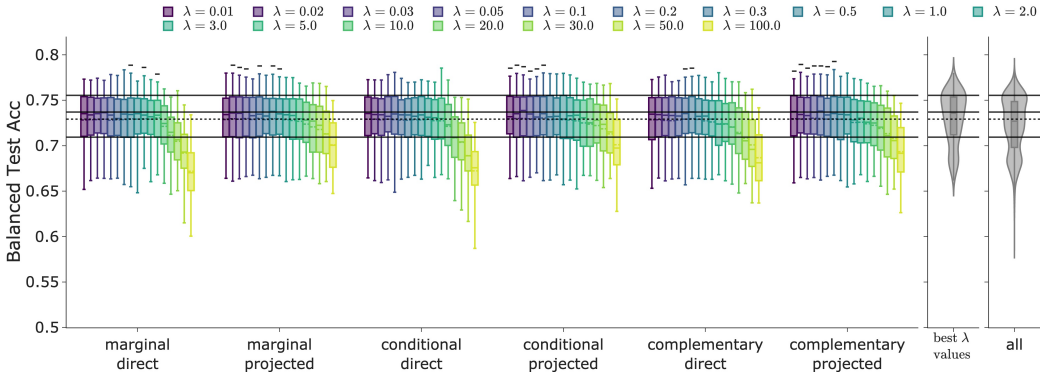
We use the two-sided Welch's t-test to compare performance between groups. Here, we find that generalization ratios for the density ratio method are significantly higher than for the adversarial baseline ($p < 1e-6$), while ratios for the Wasserstein method are significantly lower than the baseline ($p = 0.0097$). Since early stopping provides useful regularization, models have stopped training earlier, at a point when balanced train accuracy is lower. This effect is particularly strong for the density ratio method (its train accuracies are particularly reduced), resulting in improved generalization ratio. Here, we see that the model with only early stopping achieves much better generalization ratios than in the main experiments,



(a) Adversarial Censoring (Baseline)



(b) Density Ratio Censoring



(c) Wasserstein Censoring

Figure 8: Balanced test accuracy using censoring and optimal early stopping. (a): adversarial classifier baseline (Sec 2.6). (b): density ratio censoring (Sec 2.7). (c): Wasserstein censoring (Sec 2.8). Boxplots show 100 trials, varying random seed and data split. Horizontal black lines show reference performance with early stopping only. When censored model's mean exceeds this reference, a symbol annotation shows significance from a two-sided paired t-test (-, $p > 0.05$; *, $0.01 < p \leq 0.05$; †, $0.001 < p \leq 0.01$; ‡, $p \leq 0.001$). Left violin plot ('best λ values') shows pooled results from best λ of each censor mode and projection type. Right violin plot ('all') shows all data pooled. *Marginal*, *conditional*, *complementary*: censoring modes (Sec 2.2). *Projected*: projection model P_{θ_P} is used; *direct*: P_{θ_P} is omitted. λ: strength of regularization in (5).

though all three censoring methods still offer an additional benefit.

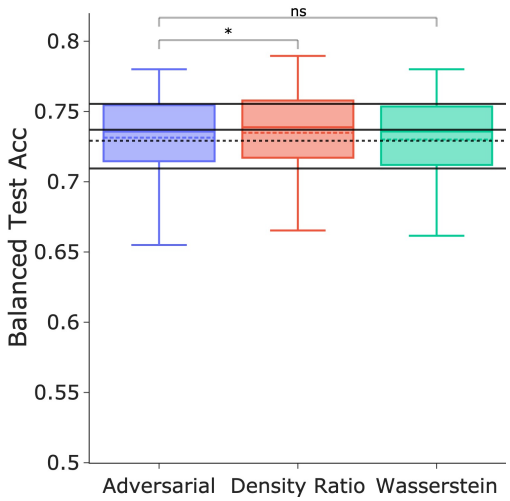


Figure 9: Peak performance across methods using censoring and optimal early stopping. Each method includes results from the λ values with best mean balanced accuracy in each censor mode and projection type (same values used in left violin plot of Fig. 8). Horizontal black lines show performance with early stopping only. Annotations show results of two-sided Welch's t-test (ns, non-significant, * $p \leq 0.05$).

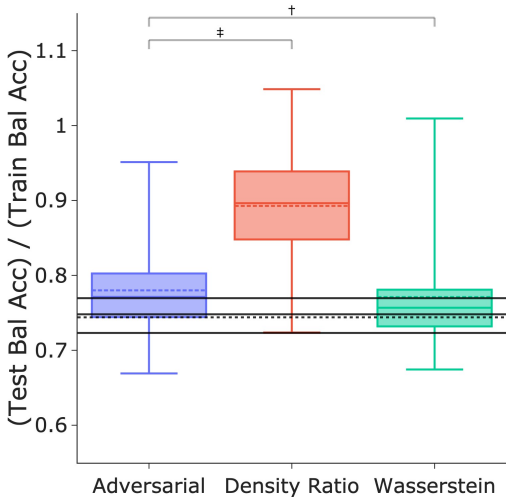


Figure 11: Generalization ratios across methods using censoring and optimal early stopping. Each method includes results from the λ values with best mean balanced accuracy in each censor mode and projection type (same values used in left violin plot of Fig. 8 and Fig. 9). Horizontal black lines show performance of model with early stopping only. Annotations show results of Welch's t-test (†, $p \leq 0.01$, ‡, $p \leq 0.001$).

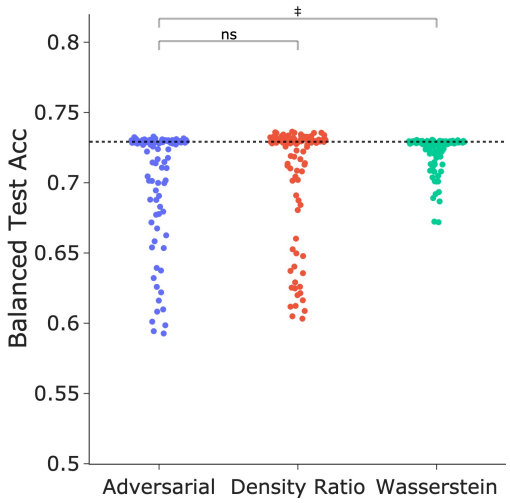


Figure 10: Stability of performance across methods using censoring and optimal early stopping. Each point is mean of a single boxplot in Fig. 8 (one censor mode, projection type, and λ value). Horizontal black line represents mean performance with only early stopping. Annotations show significance of Levene's test for unequal variance (ns, non-significant. ‡, $p \leq 0.001$).