Thinker invariance: enabling deep neural networks for BCI across more people

IOP Publishing J. Neural Eng. 17 (2020) 056008 https://doi.org/10.1088/1741-2552/abb7a7

Journal of Neural Engineering

OPEN ACCESS

RECEIVED
5 April 2020

REVISED
3 September 2020

ACCEPTED FOR PUBLICATION
11 September 2020

PUBLISHED
9 October 2020

PAPER

Thinker invariance: enabling deep neural networks for BCI across more people

Demetres Kostas[1,3] **and Frank Rudzicz**[1,2]

[1] University of Toronto, Vector Institute for Artificial Intelligence; Toronto, Canada
[2] University of Toronto, Li Ka Shing Knowledge Institute, St Michael's Hospital; Toronto Canada

Keywords: deep neural networks, BCI, brain computer interface, brain machine interface, transfer learning, domain generalization, fine-tuning

Abstract

Objective. Most deep neural networks (DNNs) used as brain computer interfaces (BCI) classifiers are rarely viable for more than one person and are relatively shallow compared to the state-of-the-art in the wider machine learning literature. The goal of this work is to frame these as a unified challenge and reconsider how *transfer learning* is used to overcome these difficulties. *Approach*. We present two variations of a holistic approach to transfer learning with DNNs for BCI that rely on a deeper network called TIDNet. Our approaches use multiple subjects for training in the interest of creating a more universal classifier that is applicable for new (unseen) subjects. The first approach is purely *subject-invariant* and the second *targets* specific subjects, without loss of generality. We use five publicly accessible datasets covering a range of tasks and compare our approaches to state-of-the-art alternatives in detail. *Main results*. We observe that TIDNet in conjunction with our training augmentations is more consistent when compared to shallower baselines, and in some cases exhibits large and significant improvements, for instance motor imagery classification improvements of over 8%. Furthermore, we show that our suggested multi-domain learning (MDL) strategy strongly outperforms simply *fine-tuned* general models when targeting *specific subjects*, while remaining more generalizable to still unseen subjects. *Significance*. TIDNet in combination with a data alignment-based training augmentation proves to be a consistent classification approach of single raw trials and can be trained even with the inclusion of corrupted trials. Our MDL strategy calls into question the intuition to fine-tune trained classifiers to new subjects, as it proves simpler and more accurate while remaining general. Furthermore, we show evidence that augmented TIDNet training makes better use of additional subjects, showing continued and greater performance improvement over shallower alternatives, indicating promise for a new subject-invariant paradigm rather than a subject-specific one.

1. Introduction

Despite the revolution that deep neural networks (DNNs) have brought to machine learning (ML) generally, they have had a more muted effect for brain computer interfaces (BCI) [1]. Despite some successes with shallow models [2–5], only limited attempts have been made to use *deeper* models with more than 2–3 layers, or modern architecture choices such as residual connections, attention, and adversarial losses [2, 6, 7]. A common claim as to why DNNs are not more consistently viable in BCI is that there are not *enough data* to train a deep model. This claim is well supported by the fact that the largest impacts DNNs have had on traditional ML have been accomplished using datasets that are *far* larger than popular public BCI datasets or data that are feasibly recorded by a single lab. Consider that the *ImageNet* dataset used to train state-of-the-art image classifiers consists of $\sim$ 14.2 *million* unique images[4], and the BERT natural language model is pre-trained with two datasets: one of 800 *million* words, and the second of 2.5 *billion* [8]. Contrast this with the summary Roy *et al* collected in their

[3] Author to whom any correspondence should be addressed.

[4] http://image-net.org/about-stats.

2019 review of deep-learning electroencephalography (EEG) analysis where, aside from a few outliers, the vast majority of datasets used in EEG research contain well under 1 million examples (despite including highly imbalanced data, and clinical data such as sleep and seizure datasets less relevant to BCI). However, there is an even more insidious challenge. Why is it that for most datasets, single-subject classification is still *more accurate* than inter-subject classification (leveraging multiple subjects) [4]? This is not accounted for by simply a lack of training examples—including more subjects dramatically increases the number of trials from which to learn; rather, this exemplifies the degree to which each subject differs. Indeed, single-subject neural-network classifiers succeed *despite* ***poorly*** *leveraging the entirety of a dataset*.

BCIs are affected by several sources of variability including equipment, subjects, sessions, and even trials (keeping all else largely constant). These differences in *context* cause differing degrees of data drift, so that classifiers *trained* in one context are suboptimal in another, even when they are meant to classify the same or similar tasks or paradigms. The resulting decrease in performance is then unsurprising; it crucially violates a fundamental axiom of *trained* classifiers and ML generally, that data encountered in practice (or in evaluation) should be independent and identically distributed (i.i.d.) with respect to the original training data. The consequences of this violation abound in practice. For example, Lawhern *et al* found with all the DNNs they considered, that when trained with the BCI Competition dataset IV 2a, a common motor imagery (MI) benchmark, classification accuracy was nearly halved for cross-subject training as compared to within-subject [3]. This was despite the fact that they matched or surpassed state-of-the-art single-subject classification using their shallow neural network *EEGNet*, and their more traditional filter-bank common spatial pattern (FBCSP) baseline.

In response, transfer learning (TL) is meant to overcome the pitfalls of limited data, and the variability between contexts by *transferring* shared qualities. Data from other *contexts* are leveraged to enhance a target context, thus better leveraging the *entirety* of relevant data available. While TL is an effective technique, there are two slightly different sub-types of TL that are often conflated in the BCI literature. We suggest that this conflation has prevented leveraging these aspects jointly, as they are not mutually exclusive. These two concepts are better disambiguated by the terms in the wider ML literature: domain adaptation (DA) and domain generalization (DG). Both of these approaches can be leveraged to enable classifiers developed for one or more subjects to be useful for one or more new, unseen people.

The differences between DA and DG lie in the treatment of the so called *target domain(s)* relative to the *source domain(s)*. For the moment, we loosely consider domains as the distribution of trials from a single subject[5], though we provide more formal definitions in section 2. DA is primarily concerned with *adapting* the trials from a set of source domains (subjects) to be *most useful* for one or more target domains (subjects). DG, on the other hand, does not target any *specific* domain, but rather transforms data from all domains in such a way as to make them most useful for *any* new relevant domain. Neither of these techniques is new to BCI, but they are commonly both referred to as TL *without distinction* and rarely, if ever, synthesized. Perhaps the most well-known examples of DA are methods that learn to select (optionally weighted) features from a pool of source subjects based on some criteria evaluated on target features. For example, the invariant common spatial patterns (iCSP) method proposed by Blankertz *et al* [9] required discovering a unique parameter ξ (optimization and cross validation were required before ξ was viable) for *each* new subject [9], with respect to the original pool of subjects; in effect, this is a simple, *target specific* transformation (mostly selection) of source domain features. Data alignment approaches such as Riemannian alignment (RA) are relevant examples of DG [10, 11]. RA ensures that all trial covariance matrices are aligned with respect to an *a priori* reference matrix and thus all trials function as features that can be leveraged for any other *aligned* trials. Crucially, the alignment is not dependent on classifiers, features, or subsequent learning for any domain-specific target; in the case of RA, alignment is performed with respect to the mean of auxiliary data from each subject to model subject-specific variability. In some work, both DG and DA approaches are compared, but they are featured in juxtaposition as mutually exclusive alternatives [12]. There is, however, no reason why a DG approach cannot be leveraged to enhance DA.

Work at the intersection of BCI, DNNs, and TL [13–17] has more commonly involved DA, typically through *fine-tuning*. This means using a pre-trained model of some sort, like a model trained jointly with a pool of subjects [13, 15]; however, others pretrain using the same subject's data [14] and, in a few cases, models used weights previously tuned for different tasks such as image recognition [18]. Using the network weights from a pretrained model, most if not all the weights are slightly adjusted (typically by using a small learning rate) to fit a *small* pool of target domain data. Some weights may be kept unchanged to minimize the shift from the original features, but the model is never expected to remain useful to non-target or even original training domains. The particular combination of DG and DNNs in the BCI literature is rare. While there are some examples of multi-*task* DG [19, 20], or larger scale data (such as clinical sleep data) self-supervision [21], the work of

[5] http://image-net.org/about-stats.

Ozdenizci *et al* is, to our knowledge, one of the only examples of generalizing across *subjects* [6]. In that work, Ozdenizci *et al* used a variational autoencoder to encode a latent space that is alternatively implicitly or explicitly generalized to be domain (subject) non-specific. This is meant to generate features of specific trials such that they are independent of subject. Unfortunately, their performance was relatively underwhelming when compared against work such as Dose *et al* [15], who used the same public MI dataset and a shallower network. While this seems to imply that DG may be superfluous, there is good reason, outlined in section 2, to suggest that their results were due to their choice of shallow architecture, one that was not sufficiently expressive.

There are gaps in both the DA and DG approaches taken with DNNs in prior work. Fine-tuning was originally designed to transfer abilities from *a large generic set* of images to a *specific* (sub)domain, for example, leveraging general natural images to *specifically* predict skin-cancer from images of skin [22]. Once transferred, there is no need for the skin-cancer classifier to distinguish between, say, lions and tigers (as may have been the case in the original data). While there is an argument for completely personalized BCI classifiers, we suggest that preserving out-of-target predictions remains valuable. As a single model's subject-specific performance is improved, it is preferable that performance is improved (or at least *unharmed*) for unseen subjects. In contrast, considering the image analogy from above: with more images of skin, we would be indifferent to performance changes in the recognition of big cats. The current gap we observe in DG amounts simply to weak resulting performance, if using DG weakens the resulting accuracy of a classifier as compared to not using it, a loss of information is the cost of generalization, appropriate DG methods should perform *no worse* than unaugmented training.

In this work, we present three contributions to *thinker-invariant* DNN training: DG-focused training augmentations that leverage non-target subjects to outperform subject-specific shallow classifiers, a multi-domain learning (MDL) procedure that leverages both DG and DA to outperform either in isolation, and an *application-appropriate* deeper DNN architecture that can make better use of the benefits of the two former contributions. These are evaluated using five separate *publicly accessible* datasets: two MI datasets (in terms of subjects, one small and one large), two rapid series visual presentation (RSVP) datasets (a visual oddball paradigm and P300 speller), and an error-related negativity (ERN) dataset. For each dataset we limit pre-processing, we perform no additional filtering unless needed to prevent aliasing, we include trials marked as 'unusable' for training, and no *dataset-specific* normalization with respect to baselines or otherwise is done. We simply crop the trials in accordance with the task and scale values between -1 and 1. We do this in an effort to demonstrate this DNN architecture and overall methodology can be easily adapted to new applications with minimal prior knowledge. To aid this further, we have made all of the code used in our work available at https://github.com/SPOClab-ca/ThinkerInvariance including both the PyTorch implementation that we used for our own experiments, and a reproduction using the more approachable Keras API as a convenience for those interested in applying this work elsewhere.

2. Background

A supervised learning problem in its most general form is taking some sort of feature space $\mathcal{X}$ and a label space $\mathcal{Y}$ and learning a function $f\colon \mathcal{X} \to \mathcal{Y}$. Any individual instance of a supervised task can be seen as $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with a probability distribution $P(x, y)$. The goal is learning f to approximate $P(y\,|\,x)$. For such problems, we can define domains as $\mathcal{D} = \{\mathcal{X}, P(x)\}$, where $P(x)$ is the posterior probability of each $x \in \mathcal{X}$ for our task $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$. Furthermore, if we focus on a BCI classifier with a set of possible subjects T that constitute our domain, we can restate our domain as $\mathcal{D} = \bigcup_{t \in T} \mathcal{D}_t$. We make the assumption that all $\mathcal{D}_t \subset \mathcal{D}$ given the common difficulty of leveraging one subject's data for another. In other words, we assume no single subject can represent the entire task domain alone.

Consider the possible sources of error that prevent an ideal determination of $P(y\,|\,x)$, which we use mostly interchangeably with $f(\cdot)$. *Empirical risk minimization* characterizes the process of learning a supervised task using a loss function l, which makes up the vast majority of modern ML, neural networks or otherwise [23, 24]. We can define *risk* for some hypothesis h of the label function f

$$R(h) = \int l(h(x), y) dP(x, y) = \mathbb{E}[l(h(x), y)]$$

Following Wang *et al* [23], we can break this risk down into two terms, considering three different hypotheses:

1. $h_{opt} = argmin_h R(h)$, the optimal hypothesis in terms of risk.
2. $h_{arch} = argmin_{h \in \mathcal{H}} R(h)$, the optimal hypothesis given the constraints imposed by the hypothesis space.
3. $h_{emp} = argmin_{h \in \mathcal{H}} R_{emp}(h)$, the hypothesis constrained by the empirical estimate of risk for a given hypothesis space.

We assume that there exists an h_{opt} that is uniquely optimal for each subject, *and* the union of all people. The error of the empirically constrained solution, with respect to the optimum, can now be expressed

as the sum of two terms: the *estimation error* and the *approximation error* [23]:

$$\begin{aligned}\mathbb{E}[R(h_{emp}) - R(h_{opt})] &= \mathbb{E}[R(h_{emp}) - R(h_{arch})] \\ &\quad + \mathbb{E}[R(h_{arch}) - R(h_{opt})] \\ &= \mathcal{E}_{est} + \mathcal{E}_{approx}\end{aligned}$$

The more expressive a model is, the greater its theoretical capacity to mimic some arbitrary function, which is accounted for by $\mathcal{E}_{approx}$. Similarly, $\mathcal{E}_{est}$ represents the better understanding captured with more or better data during training. Consider a classifier (e.g. a shallow network, or constrained kernel-based classifier) exhibiting high performance (thus low risk) within a **single** subject domain D_{ss}—which error terms are likely to be low? Under our assumption of a unique optimum, we can make some inferences about its error from the change in performance from adding new subjects. If performance remains high, both terms are already likely low. The empirical estimate of $P(x, y)$ would seem to satisfactorily cover additional domains, and the architecture satisfactorily approaches the globally relevant optimum. However, what if performance deteriorates with the addition of new subjects? There *should* be little change in $\mathcal{E}_{approx}$ if using an appropriate classifier, as if it were globally appropriate, adding more relevant data should not affect this error (*i.e.* it is not dependent on R_{emp}). If there were a change, it implies that the classifier was *only* optimal in a single domain (subject) and not globally optimal—in other words, *overfitting*. If the classifier were appropriate, then $\mathcal{E}_{est}$ has increased due to R_{emp}. DG is then an attempt to transform $g(\cdot)$ each person as domain D_t to a new invariant domain $D_{TI} \approx g(D_i), \forall i$, so that $\mathcal{E}_{est}$ is consistent across different people, and each additional subject seen during training should serve to bring h_{emp} closer to h_{arch}.

Consider that, for example, EEGNet has been highly successful on a subject-specific basis, but not when leveraging more subjects [3]. This would seem to encapsulate the challenge we describe above, so we propose to use DG to improve E_{est}, and then minimize $\mathcal{E}_{approx}$ using a potentially *more suitable hypothesis space* to find $h_{arch} \approx h_{opt}$, for example a DNN that is deeper and more expressive. This is in general supported by two important factors: the hypothesis space itself, *i.e.* the choice of architecture and the operations it can perform, and secondly, how it is searched (*i.e.* how the $argmin_h$ is performed). Crucially, some $h \in \mathcal{H}$ are easier to find than others, and thus the optimization procedure employed plays a large role [23]. As is detailed further in the sections below, to avoid biasing our results by using better optimization techniques than might not have been available to our baselines, we re-implement and train both ours and baseline models using the same optimization procedures. We address the choice of hypothesis space by presenting what we claim is an improved DNN architecture for BCI generally (where it is closer to h_{opt} than prior work).

3. Methods

3.1. TIDNet: DenseNet-inspired isolated spatio-temporal operations

The most commonly used DNNs in BCI work are the shallow convolutional network (SCN; its deep counterpart seemed to be featured less often in the work we investigated) proposed by Schirrmeister *et al* [2] and the EEGNet architecture proposed by Lawhern *et al* [3]. The common factor in these networks, and others inspired by them [15, 20, 25], is the marked separation of temporal and spatial convolutions. Our proposal is also inspired by this feature, but we offer two intertwined criticisms of the previous approaches that we address with our own: both the SCN and EEGNet architectures do not use a non-linear activation between the isolated temporal and spatial networks, but rather apply two linear operations. Secondly, the structure of these architectures provides no clear approach with which to increase the number of layers used, particularly while respecting spatiotemporal isolation. These are related criticisms in some sense as, if there were a non-linearity between the temporal and spatial operations, simply repeating either operation may be an effective strategy, which we explored in our own previous work [26], at increasing architecture depth and model complexity.

Interestingly the major drawback of the absent non-linearity is likely in fact the *advantage* of these two approaches, as this operation remains nicely constrained. While the absence of non-linear activation preserves an overall linear nature to the composition of the temporal and spatial operations, the convolution operations impose constraints. Given the apparent advantage of these isolated operations over convolution operations that span time and channels (space), these constraints are crucial. If these linear layers were simply repeated, the composition still remains of course linear, but through successive composition the overall operation is *less constrained*. In other words, simply adding more linear layers amounts to loosening the necessary constraint and enabling the sub-optimal operation that spans both time and channels (space). The challenge then is ensuring that appropriate spatiotemporal isolation constraints are preserved, while increasing the depth and complexity.

To address the concerns above, and to focus on extracting appropriate features, we construct a thinker-invariant DenseNet-inspired DNN (TIDNet). It features two clear semi-isolated stages, with operations that focus on extracting temporal and spatial features respectively, followed by a simple classification layer. Residual connections exist throughout, but crucially take two different forms depending

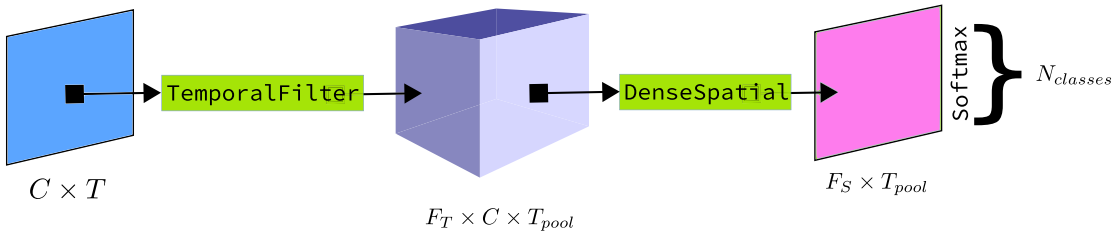

Figure 1. The overall TIDNet architecture consists of two feature extraction stages followed by a simple classification layer. Starting from an input trial on the left of C channels and T time samples, temporal filtering is first performed as detailed in figure 2, which produces T_{pool} responses of a set of F_T filters for each channel C. This is followed by a DenseFilter stage that aggregates spatial operations across the C channels by appending filters to F_T to produce F_S. This is then flattened into a vector and a single layer translates this to classification targets normalized to a probability distribution using the *Softmax* operation.

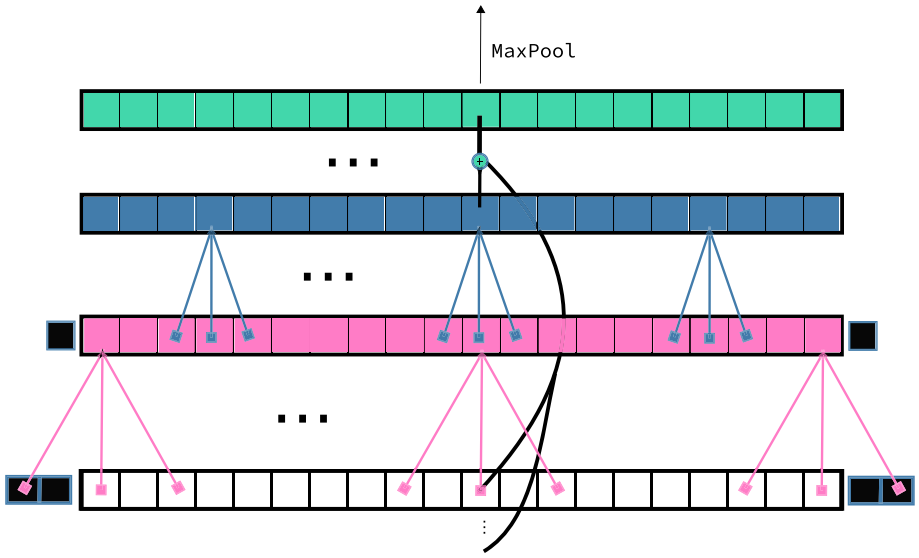

Figure 2. Temporal filtering stage in TIDNet. The bottom row vector represents one of C channels with length T. Each layer up is a convolution operation, note the dilation (skipping samples).

on the stage. Similarly, standard non-linearity operations are used throughout the architecture, in this case, the leaky rectified linear unit [27]. By design, the architecture depth can be easily extended, independently in each stage, to model more complex temporal or spatial features. Although, for consistency of dataset treatments, we do not test this feature here, opting to focus on its features *'out-of-the-box'*, any hyperparameters not detailed in the next sections can be found in Appendix A.

3.1.1. TIDNet: temporal stage

Looking specifically at the temporal filtering stage of TIDNet, figure 2 shows the time series of a single channel and two convolution layers. We show a single sequence at each layer for simplicity, but there is typically more than a single channel, and TIDNet consists of many more filters at each layer. Each convolution operation spans only along the temporal sequence of each channel, in keeping with the principle of isolation we discuss above. The convolution weights are regularized using *weight normalization* [28], and the result of each convolution operation is followed by our non-linearity and by filter-level dropout; the effect of different *filters* are removed during training [28].

The major highlight of this structure is in including both dilation and a residual connection, similar to work in convolution-based sequence modelling [28]. Dilated convolution operations are performed by skipping $d-1$ samples in the weighted sum of the convolution for a dilation rate of d. So, for a dilation rate of d and samples $s_t, t \in [0, T)$, if a convolution operation with three weights w_1, w_2, w_3 was applied, the subsequent output is $\tilde{s}_t = w_1 \cdot s_{t-d} + w_2 \cdot s_t + w_3 \cdot s_{t+d}$. Dilations are used in TIDNet to increase the span of temporal operations over subsequent layers, without needlessly increasing the number of parameters. Since many BCI datasets are normally sampled at rates well above what is necessary, and knowing that handcrafted BCI features focus on trends over *tens* of samples rather than immediate changes, we begin the earliest layer with the largest dilation rate, equal to the number of layers in the stage, and reduce the dilation rate by 1 for each subsequent layer. The last layer is then a standard

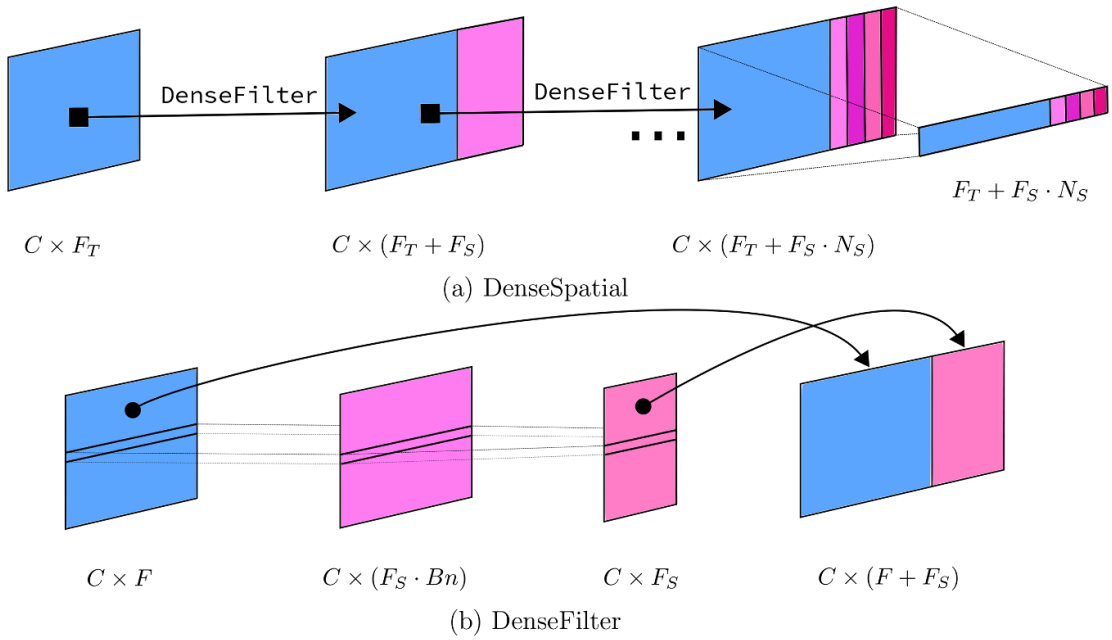

Figure 3. The spatial stage is composed of N_S layers of the *DenseSpatial* operation in (a), itself composed of F_S filter activations that are produced by the *DenseFilter* operation (b). Thus, the spatial stage consists of *concatenating* F_s new activations for each of the N_s layers to the previous layer, starting with the F_T filters from the *temporal stage*, then finally integrating the activations across all C channels. The *DenseFilter* operation in (b) consists of three main parts: a 1D convolution across the C channels with $F_s \cdot Bn$ filters (where Bn is an inflation factor to create a bottleneck), a collapsing of the inflated (by the factor Bn) filters to F_S (the desired new activations), followed finally by concatenation with the original activations. Note that a non-linearity is applied between the first and second operations within the *DenseFilter*, a dramatic increase in non-linear activations as compared to previous work, even when there are a limited number of layers.

convolution with $d = 1$ (which is in fact the opposite of what would be more commonly used in speech processing [28] or in other uses with EEG [29]). To illustrate how this dramatically increases the potential capture for long-term trends with fewer parameters, consider the number of parameters needed to roughly match the temporal span of EEGNet (64 samples in length). With only two layers and 11 weights at each layer, the final output includes samples from the original sequence between $t - 30$ through $t + 30$, but consists of only 22 convolution parameters (roughly a third of the parameters needed by EEGNet). We find through our early design process that setting the number of weights per convolution at approximately 5% of the sequence ($N_{weights} = 0.05 \cdot T$) provided a good trade-off between speed and accuracy.

While dilation allows TIDNet to capture long-term trends, a residual connection facilitates integrating short-term trends. Residual connections vary in type, but here we simply perform the weighted sum of the unmodified input series with the output of the temporal stage up to this point. This also facilitates the stage constructing features in terms of difference with respect to the input, facilitating deeper structures and more stable training [28, 30]. Finally, it is important to note the dramatic increase in dimensionality that results. Multiple convolution kernels (*i.e.* filters) at each stage requires the expansion of the original matrix to a three dimensional tensor. Including batch dimension during training, the expansion is from 3 to 4 dimensional tensor. The temporal aspect of the output of this stage is therefore max-pooled to both limit the memory requirements and regularize the features.

3.1.2. TIDNet: spatial stage

The primary goal of this stage is to transform the channel space sequences to another latent space with strong discriminatory power. Given that the most common problem utilizing an overpowered network with limited data is the tendency to overfit, care is still needed when increasing the depth of the network. As discussed in the introduction, despite leveraging the data of many people, the number of training examples still remains small relative to other classical ML fields, so the regularizing effect of many examples is less reliable. While the dilations above help reduce the tendency to overfit, they are not similarly suited to channel space data. While the sequence of C channels is *loosely* spatial, it is only this way with some immediate neighbouring elements (and not so with other neighbours). There have been different attempts at re-creating the spatial nature of channel space for training DNNs [31, 32], but these are relatively less common and do not perform quite as well as networks that use channel space directly. Instead we opt for a system that reuses the activations of local integration from previous layers multiple times.

The DenseNet [33] family of DNNs is well known for its strong performances in image classification

tasks while retaining parameter efficiency through its unique residual structure. We attempted to circumvent our overfitting concerns, *and* the poor spatial relation of our data format by simply translating the DenseNet structure to 1D convolution kernels (rather than the standard 2D image kernels). The advantage of this is that despite the depth of the network, the number of parameters remains small by the continual reuse of layers (other DNNs would need to re-learn partial transformations) [33]. An additional advantage is that gradients easily propagate back through the DenseNet network [33], potentially allowing a *short-circuit* when temporal information is of primary importance (e.g. ERP datasets).

The spatial stage then consists of multiple layers of translated *DenseNet*-like operations (figure 3a) that we called DenseFilters, followed by a final channel mixing step. The goal is to iteratively expand the *filter space* into a set of strong local features that are augmented by progressively global ones, and then remove the remaining effects of *channel space*, to ensure that the former has expressed the needed variation of the latter. Each DenseFilter (figure 3b) is composed of two layers, each consisting of a LeakyReLU, batch normalization [34], and convolution operation. The first layer creates a bottleneck—a 1x1 convolution (filter space transform) that expands the number of filters as a composition of incoming filter activations. The second layer, with activations from the expanded/-composed filter set performs spatially isolated convolutions with a kernel span of 5 in channel space (1 in temporal space for isolation), selected for simplicity. The activations of the resulting filters are then concatenated with the original incoming filters.

3.2. Euclidean alignment

Given our framing of DG as minimizing the empirical risk estimate of a single invariant domain rather than a set of possible subject domains, a simple methodology follows. We leverage characteristics consistent across all domains and try to express instances with respect to this reference. *Euclidean alignment* (EA) [10] is a data alignment technique that can be seen as narrowing the scope of $P(X)$, and consequently $P(y, x)$. As a DG strategy, it follows the methodology of projecting data into a domain-invariant space, leveraging the fact that EEG is a continuous multivariate time series, implying it has a symmetric positive definite covariance matrix. It takes its inspiration from RA, which shifts each trial for each subject with respect to the Riemannian mean of a set of *resting* trials, but instead trials under EA are expressed simply with respect to the identity matrix. While RA asserts perhaps a stronger prior, and thus may be more effective [1, 35], it requires the subsequent use of Riemannian classifiers, which operate on covariance matrices in Riemannian space rather than raw time-series. Although, previous work attempted to perform manifold learning with DNNs generally [36], the vast majority of DNN research is still tailored towards learning in Euclidean space. The advantage of *EA* is that operations are performed exclusively in Euclidean space and thus do not exclude standard DNN approaches. The alignment process of EA is centred around asserting a consistent mean covariance matrix of *all the trials* for each subject—in this case, the identity matrix. Consider a subject with n trials each represented by matrix M:

$$\bar{R} = \frac{1}{n}\sum_{i=0}^{n} M_i M_i^T$$

Alignment is then simply performed by using the matrix square root of the arithmetic mean:

$$\tilde{M}_i = \bar{R}^{-\frac{1}{2}} M_i$$

After re-weighting M_i, the mean of all covariance matrices becomes the identity matrix I:

$$\begin{aligned}\frac{1}{n}\sum_{i=0}^{n} M_i M_i^T &= \frac{1}{n}\sum_{i=0}^{n} \bar{R}^{-\frac{1}{2}} M_i M_i^T \bar{R}^{-\frac{1}{2}} \\ &= \bar{R}^{-\frac{1}{2}} \Big(\frac{1}{n}\sum_{i=0}^{n} M_i M_i^T\Big) \bar{R}^{-\frac{1}{2}} \\ &= \bar{R}^{-\frac{1}{2}} \bar{R} \bar{R}^{-\frac{1}{2}} = I\end{aligned}$$

Note that, while data for the subject is needed to establish the unaligned mean, labels are *not* needed. In the original presentation of EA, He *et al* visualized the pre- and post-alignment distributions of trials. The alignment procedure created much more homogeneous and less sparsely distributed data for each subject ($P(x)$). This suggests that it requires less data to achieve an accurate empirical estimate of the data distribution, as it is isolated within a smaller subspace. For convenience, alignment was computed for (and applied to the trials within) each single recorded *file* for each single subject. Finally, this process is notably similar in methodology as that of the more conventional process of spatial whitening [37], where all trials are first concatenated and then a global covariance is used to align data, instead of the mean covariance across trials.

3.3. Mixup

The second approach we consider is *mixup* regularization. This technique has the effect of enforcing more linear model behaviour in the immediate vicinity around training points [24], the degree of this is tuned by a hyperparameter α. It can also be interpreted as a *data-augmentation* technique as, in effect, linear interpolation is used to create artificial points between training points. However, it was originally motivated from the standpoint of improving the empirical estimate of $P(y, x)$. It trades the naive empirical estimate, which is a probability distribution

$P(x, y)$ made up of the sum of Dirac masses located at each training point (x_i, y_i), with a *vicinal* estimate that assumes a *vicinity distribution* ν around each point. The probability of finding the *virtual* point $(\tilde{x}, \tilde{y})$ is:

$$P(\tilde{x}, \tilde{y}) = \nu(\tilde{x}, \tilde{y} | x_i, y_i)$$

Thus, mixup is an assumed prior distribution for the empirical learning problem that functions as a stochastic interpolation between existing points, driven by the parameters $\lambda \sim Beta(\alpha, \alpha)$ and $\alpha \in (0, \infty)$:

$$\nu_{mixup}(\tilde{x}, \tilde{y}) = \frac{1}{n} \sum_{j=0}^{n} \mathbb{E}[\delta(\tilde{x} = \lambda \cdot x_i + (1 - \lambda) \cdot x_j, \tilde{y} = \lambda \cdot y_i + (1 - \lambda) \cdot y_j)]$$

Furthermore, Zhang *et al* suggested another potentially relevant feature of mixup that could benefit the approach we take here. As we are particularly interested in employing a relatively *deep* and computationally expressive neural network, it becomes easier for the network to fall into the degenerate case of memorizing training points [24]. Since mixup constantly presents a modified version (the degree of this is tuned by increasing α, if $\alpha = 0$ we never present virtual points) of the original points, the *exact* values of the points become much more difficult to memorize outright [24], minimizing this risk when using a network with a greater capacity for memorization.

3.4. MDL as DG-DA fusion

The distinction between what is strictly DG and MDL requires some consideration. Recall the 'thinker-invariant domain' from above, $\mathcal{D}_{TI} = \{\bar{\mathcal{X}}, P(\bar{X})\}$. DG methods attempt to produce a single function $f_{TI} : \bar{\mathcal{X}} \rightarrow \mathcal{Y}$ independent of source domain. MDL on the other hand constructs a similar f_{TI}, but through a slightly different process. At least latent in a MDL model is some identification of source domain d_i, through a factor z implicitly $(z = g(x))$ or explicitly $(z = d_i)$, and f_{TI} is approximating $P(y|\bar{x}, z)$. We can see MDL and DG along a continuum then, at one end $z \approx d_i$ and the learning problem is MDL, at the other $z \not\approx d_i$ and the problem is DG. A strict implementation of DG might *penalize* any $z \approx d$ at different layers. We aim not to penalize any ability to distinguish domain, which has been tried before with less success, for example by using a conditional variational auto-encoder (cVAE) to remove domain as a variable, or adversarial penalties to penalize identification [6]. These methods seem too harsh a restriction, preventing the internal ensemble that may require some sort of domain identification. In practice, we suspect that most forms of DG are to differing degrees still MDL; for example, Ozdenizci *et al* found that subject identity could still be determined by their cVAE's latent parameter z well above chance levels, albeit at a dramatically reduced rate to not using the cVAE [6].

Interestingly, if we accept that *practically* most expressions of DG are bound to be MDL to differing degrees, a further possible training strategy becomes apparent. Recall that the closer the original training domain is to the test/online domain, the better the quality of training data by the i.i.d. principle. Although a classifier trained with some reserve set of subjects that requires no tuning in the future is best in online environments, it is not uncommon or particularly difficult to have *some* training data from a new subject. The common motivating premise of TL was to positively leverage reserve data to make use of *minimal* target data. Our proposal is to re-frame this in terms of MDL, rather than DA or DG, by simply including a small amount of training data from the target domain to improve performance for that subject. This is a known and simple approach considered in the TL literature, that often amounts to similar performance to much more complex methods, using largely the same data, all while being faster and relatively simple to implement [38].

3.5. Datasets

All datasets are publicly accessible, and are selected to represent tasks better separated spatially and more temporal evoked potential tasks: two MI tasks followed by a P300 speller and ERN (itself from a P300 speller's feedback) respectively. We supplemented these four datasets with a visual oddball task used by the authors of the EA technique for a comparison focused on EA. We use PhysioNet [39] and prioritize *larger* datasets, in terms of the number of people, where possible. We particularly select for those with existing DNN benchmarks or, in the case of the P300 speller, a performance level using DNNs that could be used for comparison. Herein, we discuss details of the tasks and datasets and briefly elaborate on related work and their previous performances to motivate our choice of reference classifier for each dataset. When there were explicit test splits (IV-2a and ERN), test performance is reported using exclusively these.

With the exclusion of the MDL experiments (see sections 3.4 and 3.6.1), all experiments are performed in a leave-one- or leave-multiple-subjects-out (LOSO and LMSO respectively) framework. This means that, for N subjects in a LOSO framework, $N - 2$ are selected as training subjects and the remaining two are held out as validation and test subjects respectively (with the test split specifically used for that subject in the IV-2a case below). For LOSO, this is repeated N times, leaving each subject out as test in turn, and selecting the previous test subject as the new validation subject. Under the LMSO framework, a similar procedure is performed for N_f folds of data, where each fold contains an even division of the number of subjects, with the first folds inflated whenever a remainder is present after equal division. So, for S subjects, each fold has $\lfloor S/N_f \rfloor$ and the first (S mod N_f) folds have one extra subject in each fold. Subjects

are selected for folds in alphanumeric order. As an example, consider ten subjects labelled S1 though S10 being split into five folds, S1 and S2 are assigned to the first, S3 and S4 to the next, and so on. If instead, there were an eleventh subject S11, they would be part of the final fold along with S10 and the first fold would hold S1, S2 and S3 (the remaining folds would hold each next sequential pair). Performance is evaluated using accuracy, balanced accuracy, and area under the receiver operating characteristic (AUROC) depending on the most common metrics used in prior work as detailed in section 3.5. We compare the performance of TIDNet to a re-implemented baseline from the literature with no DG, with EA, with mixup, and with both EA and mixup. In other words, we compare TIDNet with or without DG augmentations to both an equivalently augmented and non-augmented baseline models. A paired Wilcoxon test across the test performance of each subject finds statistically detectable differences in performance.

3.5.1. BCI Competition IV Dataset 2a (IV-2a)

These recordings are of a four-class MI dataset originally recorded to represent a continuous multi-class challenge for session to session transfer, as a part of the BCI competition IV [40]. Nine subjects performed imaginary movement of the left hand, right hand, both feet, and tongue in equal proportions, with six runs of 48 trials each. Each class is represented 12 times per run, repeated across two sessions per subject. The sessions are crucially performed on *different days*, to challenge classifier transfer of capability. Twenty-two channels were used to record EEG signals and three for electro-ocular (EOG) signals. All were sampled at 250 Hz, with a band-pass filter between 0.5 and 100 Hz and 50 Hz notch filter to reject line noise.

This dataset has been consistently, and well utilized by many people using DNNs for BCI work over the last several years [2, 3, 41], most of which follow in some way or another the known pattern of isolated temporal and spatial stages, with mostly linear layer activations as described in section 3.1. Lawhern *et al* in particular considered cross-subject training, but notably leveraged only five subjects of nine when training their general model. This was perhaps not enough data, and we further investigate these requirements with the larger MI dataset, in section 3.7. This makes it a poor direct comparison to our use of at least eight subjects. The authors compared their own EEGNet and Schirrmeister *et al*'s shallow and deep models, and found no discernible difference in performance across models, achieving approximately 40% 4-way classification accuracy. All of these works *further* band-pass filtered their data between 4 and 40 Hz, and used a z-score-like normalization procedure based on the exponentially weighted mean and variance as described by Schirrmeister *et al* [2]. As mentioned in section 3.6, we avoid these dataset-specific pre-processing steps where possible, but as multiple prior works leveraged the highly cited ShallowConvnet (SCN) model, and this dataset features in the original presentation of this model, we elect to use this as our baseline for this dataset, and use the author's own implementation found at . We similarly use the 4.5 s time period from $[-0.5\text{ s}, 4\text{ s})$ around each event marker, and use classification accuracy as our evaluation metric.

This dataset also elucidates suitable hyperparameter selection for TIDNet and mixup across all datasets. Rather than performing searches for all datasets, we create an out-of-box solution based on strong hyperparameters for this dataset. To do this, we performed one hundred iterations of a random hyperparameter search, maximizing the validation accuracy of the first fold of our LOSO training. This causes our parameters to be particularly well suited to this validation fold (and dataset to some degree), which for our implementation of LOSO training implies potential overfitting to the test performance of the final subject. While we notice no obvious difference in this subject's performance compared to the remainder, the current configuration of TIDNet is particularly well suited to this dataset at the very least. To avoid strong errors, we exclude this subject from the statistical test for this dataset, and used only the remaining eight subjects.

3.5.2. Physionet Movement and Motor Imagery Database (MMI)

This dataset [42] consists of 109 subjects, where each performed 14 experimental runs. The first two runs are baselines, and the subsequent twelve are three repetitions of four tasks. The first and third are the *not-imagined* counterparts of the second and third, respectively, and we ignore them for our own work. The second task consisted of two possible sub tasks: the imagined opening and closing of either the left or right fist respectively, and the fourth run, rather than the left or right fists, consisted of imagined opening and closing of *both* fists or feet. The recordings were made with 64 EEG channels sampled at 160 Hz, in a 10-10 montage with exceptions of electrodes Nz, F9, F10, FT9, FT10, A1, A2, TP9, TP10, P9, and P10.

Much like IV-2a, MMI has been used for previous experiments that use DNN classifiers [6, 15]. As we outlined in the introduction above, Ozdenizci *et al* [6] leveraged a thinker-invariant DG approach with this dataset. Their work heavily relied on using the original SCN architecture, so that they could focus on DG. While they successfully made a more general model, it was less performant than the work of Dose *et al*, who use a very similar architecture but managed absolute accuracies on the order of 10% higher without DG. Dose *et al* [15], for the most part, used the SCN model, but with the more conventional rectified linear (ReLU) non-linear activation [27],

rather than the FBCSP-like activations. Furthermore, rather than focusing on DG, Dose *et al* used fine-tuning (DA) and further boosted the absolute subject-specific classification accuracies by 6.11%, 9.43%, and 9.92% for their two, three, and four-class problems they defined (which we also use, below) respectively. Given that Dose *et al*'s performance was well above even the SCN implementation alone (Ozdenizci *et al* also trained a baseline without DG), we use this model as our baseline until, during initial experimentation, it became difficult to achieve anything but chance level performance using this model in conjunction with EA . We instead try the EEGNet model, and find that its performance meets or surpasses the performance of the architecture used by Dose *et al*.

When examining the dataset, we find that, in the copy of the data we had access to, subjects S088, S090, S092, and S100 contained unusable data, and are therefore removed, leaving 105 usable subjects. These are trained in a LMSO fashion using five folds to accord with Dose *et al* [15]. Again, for ease of comparison with Dose *et al*, we also turn the two binary imaginary tasks into three different problems: a two-class comparison that distinguishes left- or right-fist from task 2, a three-class problem that adds segments of the eyes-open baseline (experiment run 1) to the two-class problem for three targets, and finally a four-class problem that further integrates the imagined open and closing of both feet from task 4 (this is appended to the three class problem for four total targets). While Dose *et al* added in random trial length segments from the baseline, we instead use N evenly spaced out segments at intervals that evenly divide the entire baseline duration. In other words, for a baseline recording of length T, we took trial lengths at the offsets $0, \frac{N}{T}, \ldots \frac{T-N}{T}$, to encourage reproducibility. Finally, we use the same 3 s time window (although, similar to Dose *et al*, it seems that the 6 s window increases performance) [0 s, 3 s).

3.5.3. Matrix Speller (P300)

The traditional matrix speller is a commonly used and important BCI paradigm, and as such, performance improvements stand to make a large impact. These data were originally recorded to better understand the limitations of the paradigm and to better model the progressive changes in the marked P300 ERP that the speller is designed around [43]. The idea behind the matrix speller BCI is to rapidly flash the columns and rows of a matrix of characters, while the subject focuses on the single character they would like to use to spell out a word. The momentary visual flash of the target character is used to elicit a P300 ERP from the subject, and the current column and row are used to localize the character. The authors followed a very common setup, using a $6\times$ 6 character matrix, and randomly flashed each column and row 20 times without replacement for 100 ms, and left a gap of 50 ms between flashes (150 ms SOA). In their work, the subjects were also asked to count the number of times the character focused on was flashed, and all but the first two subjects were also asked to report this number for recording. There were a total of 12 subjects, from which we excluded subjects s08, s10, and s12 due to missing annotations, for 9 total subjects for LOSO training. The recordings were made using 64 EEG channels, four EOG channels (horizontal and vertical for both left and right), and two earlobe references for a total of 70 total time series, each sampled at 2048 Hz. We used all 70 channels, but unlike the rest of our work, opted to perform some down-sampling due to the coupling of higher channel numbers and sampling rate limiting compute capabilities. To adhere to our premise of minimal to no preprocessing, we first low pass filtered the data at 120 Hz (zero phase finite impulse response using hamming window) to prevent aliasing, and then simply decimated the data by two. In other words, we took every other sample for a new sampling rate of 1024 Hz. We take a trial window of $[-0.05\text{ s}, 0.65\text{ s})$, including pre-trial data for self-baseline-referencing. Unlike the other datasets considered, this dataset also provided left and right earlobe measurements in addition to EEG (and EOG)

Interestingly, prior work applying DNNs to P300 signals showed less consensus on architecture choice. Recognizing the need here to capture a temporal signature is likely crucial, and therefore it is no surprise that recurrent neural network architectures feature more commonly [19, 31, 44] than with MI tasks, given their propensity for sequence modelling (although they find themselves presently out of fashion for this purpose). That said, shallow separated (mostly linear) temporal and spatial convolution architectures were also well represented [3, 31, 45–47]. Throughout most of the work we encountered, aside from the work of Ditthapron *et al*, we find consistent use of P300 datasets recorded for BCI competitions II and III[6], which feature only one and two subjects respectively, making them poor choices for thinker-invariant evaluation. Additionally, as these feature so few subjects, it is perhaps no surprise that shallow approaches remain dominant. The work of Ditthapron *et al* represents a clearly relevant approach to ours, as they employ DG across P300 *tasks*. The authors used a 7-layer encoder (6 layers decoding, 1 classifying the latent encoding) to create a multi-task latent encoding. They created general and fine-tuned models in a leave-one-*task*-out procedure. Crucially, they feature the same P300 dataset that we use, and thus would appear a good baseline for comparison, but their approach required creating a latent space common to *multiple* recording contexts, out of the scope of our current focus on differences due to subjects, but we use their results for reference. As Ditthapron *et al*

[6] http://www.bbci.de/competition/iii/.

used EEGNet as an additional baseline model, and Lawhern *et al* [3] reported results using EEGNet with their own P300 dataset, we select EEGNet as our reference model.

Finally, unlike the MI datasets above which have balanced classes, the P300 dataset is distinguished by a highly imbalanced class ratio (in this case 5:1, see 3.6 for how training was adjusted to compensate) needed to evoke the P300 response [48]. There are many different potential metrics for reporting performance despite imbalances; we use the AUROC common to both prior uses of EEGNet above.

3.5.4. Error related negativity (ERN)

Pairing nicely with the previous dataset, these data were originally collected for the 2015 BCI Challenge Kaggle competition[7], with the intent of detecting when an error was made by a P300 speller. The speller system would provide feedback about which letter was selected, and the EEG recordings of this feedback were used to determine the likelihood of mistakes. While this paradigm does not appear to have any particular DNN architecture lineage like the MI or P300 paradigms, there are existing benchmarks of performance [3, 20]. Once again, EEGNet is a relevant reference point, as Lawhern *et al* used this dataset in their original publication [3], showing competitive performance to the entries of the original competition.

This dataset features 26 subjects, with 16 explicit training subjects and likewise 10 for testing. For comparison with the original competition numbers and Lawhern *et al*, we modify our LMSO procedure to use the ten reserved test subjects, and split the remaining training data into four folds for cross-validation: three for training and one for validation. The data were composed of 56 EEG channels in a standard 10-20 arrangement recorded at 600 Hz, which we leave unchanged for training. As trial window, we select $[-0.5\text{ s}, 1.5\text{ s})$.

3.5.5. Visual oddball RSVP (N2PC)

This dataset is included to provide an ERP performance reference (IV-2a was also used) to the original work presenting EA. To our knowledge, it has not previously been considered by any work using DNNs, so rather than use a reference DNN, we simply compare the subject-wise performance originally reported by He *et al* [10]. While this dataset was originally collected to facilitate the more difficult task of approximately locating an odd image change (a plane added to an overhead view landscape), it is fundamentally an oddball RSVP task which results in a P300 response [48] that can be detected, and as is used by He *et al* [10].

These data consist of 11 subjects presented images at three different rates 5, 6 and 10 Hz, of which we use the 5z to match He *et al* [10]. The number of available electrodes were limited to only eight EEG channels (PO8, PO7, PO3, PO4, P7, P8, O1 and O2), each sampled at 2048 Hz and band-passed by the original investigators between 0.15 and 28 Hz. As such, we felt no need to unnecessarily keep such a high sampling rate and needlessly boosting the number of parameters, so the data was decimated by 4 (took every 4*th* sample). Finally, we took a trial windows of $[-0.05\text{ s}, 0.65\text{ s})$ for consistency with the P300 dataset.

3.6. Model training and evaluation

We train each model using the Adam optimizer with the AMSGrad fix [49], and we used a cosine learning rate policy with warmup. This means the rate would peak at a default value of 0.001 after a linear warm-up period which lasted one fifth of the total epochs, and then a cosine decay for the remainder of training, which has been beneficial in many instances of deep network training [50]. The learning rate at any epoch e of total epochs E is:

$$\eta(e) = \begin{cases} \eta_{\max} \times \frac{5e}{E} & e < \frac{E}{5} \\ \eta_{\max} \times \frac{1}{2}\left(1 + \cos\left(\pi \frac{(e - \frac{E}{5})}{\frac{E}{4}}\right)\right) & e \geq \frac{E}{5} \end{cases}$$

We include a second-order weight decay (L2) of 0.01 across all network parameters and dropout throughout the TIDNet is 0.4. We also employ label smoothing as implemented by He *et al* [50] at a factor of 0.2 to penalize overly confident predictions, and model some noise in the labels. As done by Zhang *et al* in the original work on mixup, we first gather batches and apply mixup within each minibatch. Specifically, a minibatch is first accumulated with a set of trials, and then we apply mixup to each trial by matching it with another from the minibatch with replacement (this could lead to scenarios where a batch is matched with itself). The number of total epochs, batch size, α_{mixup} and number of final epochs to smooth weights over are the only parameters that vary between datasets (and they do not vary much), determined heuristically by finding values that ensured *validation accuracy* mostly increments over the first training fold/subject of the respective datasets. A summary of these hyperparameters for each dataset can be found in Appendix A. Wherever there was an imbalance in examples between classes, we under-sampled the majority class(es) without replacement so that the total samples drawn is equal to the number of classes multiplied by the number of examples in the minority class. Furthermore, as hinted above, to stabilize the final model weights, we save the parameters from the last N_{ewma} epochs and sum all the exponentially weighted parameters for each epoch e by $w^e_{ewma} = c_e \cdot 0.5^{N_{epochs}-e}$, where $c_e = w^e_{ewma}(\sum_i^{N_{ewma}} w^i_{ewma})^{-1}$).

Electro-ocular channels are ignored for the three datasets where they are provided (IV-2a, P300 and

[7] https://www.kaggle.com/c/inria-bci-challenge.

ERN) for consistency with the other datasets which are only available this way (see Appendix B for an ablation on the effects of EOG on TIDNet and baselines). The *A1* and *A2* reference electrodes from the P300 dataset are *included* in training to allow for learned referencing, see the end of Appendix B for results that also exclude these references. Secondly, we also include those trials that were originally annotated as bad, meaning predominantly artifacts or unusable. While these are normally rejected to increase classifier performance, in our experience with larger models, these additional trials do not hamper training. Filtering is limited to that performed by the original recording equipment or data providers, and downsampling is only used in the P300 and N2PC datasets due to their much higher sampling rates. These are detailed in section 3.5. Each trial is normalized to range between -1 and 1, preserving relative scale between channels (trial values are divided by the maximum absolute value per trial), but no baseline subtraction is performed. Thus, any dataset-specific preprocessing that may be necessary is left to TIDNet to perform internally, as our approach aspires to be as much of a self-contained and end-to-end solution as possible. While we do not explicitly evaluate this, any specialized filtering or normalization that does increase the downstream signal to noise ratio should provide similar boosts in performance to *any* selected classifier, save for *coincidental* overfitting (where the pre-processing is likely not affecting signal to noise ratio at all).

3.6.1. Multi-domain learning

For our implementations of MDL, we reserve target data by taking the first of multiple runs if more than one and fewer than four runs are performed. Otherwise, we select a quarter of the test dataset, selecting the first $n_c \times p_c$ points, where n_c is the number of points in that class and p_c is the probability of that class observed for that subject. This is done for both validation and test subjects, so that the validation domain is an accurate simulation of test conditions, albeit with different subjects. Furthermore, in LMSO experiments, the above is performed for *each* test and validation subject in their respective folds. Thus, every MDL experiment has data from every subject for each respective dataset. We make the same statistical comparisons to baseline models under MDL evaluation.

We also compare how effective MDL is in comparison to fine-tuning the general model to specific individual subjects, as was done in previous work [4, 15], and as is common in other ML fields. Fine-tuning DNN parameters (or, at the very least, re-training the final classification layers) is generally a simple yet effective method for TL [51]. What is less often considered in the context of BCI is the potentially catastrophic effect on generality after multiple training stages [52]. We expected that a MDL approach would remain effective on unseen subjects while fine-tuning would not—a potential advantage if fine-tuning and MDL target performance are largely similar in performance. To this end, we focus only on the two datasets that we trained in the LMSO scenario, providing a test-bench subset of four tasks, three from the MMI dataset and one from the ERN. The reason we focus on LMSO datasets is to study the performance of **both** target subjects *and* unseen subjects. A subject-wise paired Wilcoxon test again identifies statistically detectable differences between fine-tuning and MDL for each task and either target or unseen subjects.

Focusing on the best performing LMSO configuration TIDNet (with EA and/or Mixup added), we *target* each subject in each test fold. To fine-tune a target-specific model, we refine the *general* model in two phases, using 50% of target data. The first phase trains a new classification layer over ten epochs, keeping all other parameters frozen. This means only the final layer's weights are updated after each batch. Then, the *entire* model, including previously frozen parameters and the new classification layer, is updated for five epochs using a tenth of the original learning rate. Both phases again use a cosine learning rate decay but no warmup. This is compared against a *limited* MDL approach, which consists of augmenting the training data (those that produced the general model above) with the same 50% of the *target* subject's data. The training procedure remains the same as for the general model. Each of these approaches results in a single model tuned for a single subject, using the *same* training data cumulatively. We then compare the performance of each of these models on the remaining *target* data and the *remaining subjects in each of their respective test folds*.

3.7. Number of subjects regression

In order to get a sense of how many people's data might be required to observe a benefit in deeper approaches, and the effect of our proposed methods on this choice, we limit the number of subjects to a subset from each fold and repeat LMSO cross-validation, in this case with ten-folds for a greater range of total accessible people. We then plot the performance of the MMI dataset against the number of people used. The cross validation was performed for each of 20 logarithmically increasing subsets of subjects, starting with 1 to a maximum of 85, for a total of 400 trained models per configuration (architecture + TI). This is repeated for four configurations, for each of the three tasks of the MMI dataset. The configurations are both TIDNet and EEGNet with no thinker-invariant techniques, in addition to the highest performing thinker-invariant configuration (EA and/or mixup) for the TIDNet and EEGNet from the base classification experiments above.

Table 1. The results of re-training the reference implementations using our more updated optimization procedures (and reprinting of EA result [10] for N2PC, which is simply an EA configuration) without and with MDL respectively.

Dataset		IV-2a	MMI			P300	ERN	N2PC
		Accuracy						
Metric		(4-way)	(2-way)	(3-way)	(4-way)	AUROC		BAC
LO/MSO	Base	59.76%	80.93%	72.39%	61.53%	0.820	0.702	–
	+ Mixup	60.55%	81.49%	71.74%	59.01%	0.816	0.712	–
	+ EA	64.31%	82.09%	73.93%	63.95%	0.819	0.724	68.80%
	+ Both	64.62%	82.16%	73.31%	62.62%	0.817	0.724	–
LO/MSO + MDL	Base	71.95%	81.69%	73.62%	62.33%	0.857	0.810	–
	+ Mixup	69.10%	82.93%	73.09%	61.33%	0.855	0.799	–
	+ EA	77.74%	82.71%	75.05%	65.03%	0.857	0.797	–
	+ Both	77.39%	82.84%	74.44%	64.04%	0.854	0.794	–

Table 2. The absolute increases in TIDNet performance metrics across examined datasets with respect to reference network LOSO or LMSO performance with *no EA or mixup used*. Values with a leading * indicate statistically detectable differences between configuration and baseline without DG, using a paired Wilcoxon test ($p < 0.05$). Bold values are significant after Bonferroni correction for all 96 comparisons (including those of tables 2, 3 and 4; significance $p < 0.000\,52$). Dashed lines under N2PC indicate that the configuration scored at chance level (0.5 BAC).

Dataset		IV-2a	MMI			P300	ERN	N2PC
		Accuracy						
Metric		(4-way)	(2-way)	(3-way)	(4-way)	AUROC		BAC
LO/MSO	TIDNet	−1.47	1.16	**3.25**	**4.56**	0.003	**−0.038**	–
	+ Mixup	−2.78	*1.09	2.16	1.48	0.002	**−0.060**	–
	+ EA	5.63	**−0.46**	**4.02**	**6.05**	0.002	**0.042**	−1.23
	+ Both	5.48	0.98	**2.60**	**4.19**	0.003	**0.046**	0.12
LO/MSO + MDL	TIDNet	−5.25*	1.06	4.51	**6.01**	−0.001	**−0.032**	–
	+ Mixup	−2.16	1.83	2.63	*3.41	0.003	**−0.058**	–
	+ EA	7.10*	**−0.37**	**6.53**	**8.91**	0.000	**0.007**	−1.52
	+ Both	7.45*	1.86	**5.31**	**6.53**	0.000	**0.007**	2.78

Table 3. The absolute increases of TIDNet over reference network in performance metrics across examined datasets with respect to the *same* augmentations, EA and/or mixup. Values with a trailing * indicate statistically detectable differences between configuration and baseline with the same DG technique, using a paired Wilcoxon test ($p < 0.05$). Bold values are significant after Bonferroni correction for all 96 comparisons (including those of tables 2, 3 and 4; significance $p < 0.000\,52$). Note that N2PC is not in this table as the baseline was simply with EA and thus table 2 is already with respect to this (and no comparison of TIDNet + mixup to Base + mixup is possible).

Dataset		IV-2a	MMI			P300	ERN
		Accuracy					
Metric		(4-way)	(2-way)	(3-way)	(4-way)	AUROC	
LO/MSO	+ Mixup	−2.78	0.53	2.81	*4.00	0.006	**−0.070**
	+ EA	5.63	**−1.63**	2.48	**3.63**	0.004	**0.020**
	+ Both	5.48	−0.26	**1.68**	**3.10**	0.006	**0.024**
LO/MSO + MDL	+ Mixup	0.69	0.59	*3.16	*4.42	0.005	**−0.046**
	+ EA	1.31	**−1.40**	*5.10	**6.21**	0.000	**0.020**
	+ Both	2.01	0.71	**4.49**	**4.82**	0.002	**0.023**

Table 4. Fine-tuning against our proposed alternative MDL in targeted performance versus general performance on unseen subjects. In our tasks, improvement is universal, with a Wilcoxon paired (person to person) comparison well below $p < 0.000\,52$ (Bonferroni correction) for all comparisons except for 2-way MMI and ERN target subject comparisons.

	Target Subject Performance		Unseen Subject Generalization	
Dataset	Fine-Tuned	MDL	Fine-Tuned	MDL
MMI (2-way)	88.19%	88.87%	73.60%	82.60%
MMI (3-way)	83.62%	86.40%	60.47%	76.64%
MMI (4-way)	76.01%	78.73%	47.67%	67.60%
ERN (AUROC)	0.828	0.868	0.626	0.743

4. Results

Table 2 captures the average **absolute** improvement on the selected metrics for each dataset in section 3.5 with respect to the re-implemented reference models (performances from table 1), and table 3 shows the same when equivalently augmenting the reference model training with EA and Mixup. While TIDNet alone or with *only* mixup shows mixed results, once alignment is included there is a nearly universal performance benefit to the deeper network (TIDNet), this improvement is much more consistent when both mixup and EA are used, with only a single (statistically insignificant) case showing very mildly worse performance. While the more temporally discriminative datasets, such as the P300 and ERN, show relatively mild improvements overall, there are more dramatic performance benefits for both the IV-2a and MMI datasets, in some cases 7%–8% *absolute* increase in classification accuracy.

The MMI results show a strong increase over prior work, particularly when considering the 3- and 4-way comparisons. Interestingly, in all three tasks, our baseline outperformed what was to our knowledge the previous state-of-the-art [15]. Dose *et al* showed 80.38%, 69.82% and 58.58% for the three tasks respectively, whereas our EEGNet baseline achieved 80.93%, 72.39% and 61.53% (our re-implementation of the model by Dose *et al* was 81.42%, 71.76% and 48.93%, but was not stable training with EA or mixup, often degenerating into chance performance, we however are more suspicious of the model architecture than anything else for this). TIDNet alone outperformed all of these, with the 3 and 4-way comparison statistically significant. After including augmentations, in each task we significantly surpass state-of-the-art. Focusing on the largest increase in performance, we find a significant 8.91% increase in 4-way classification performance using EA, jumping from 61.53% to 70.44%.

Looking at the IV-2a dataset alone, results are more mixed as previous work by He *et al* used this dataset in their original presentation of EA and scored an average LOSO accuracy of 73.53% [10] using a more traditional set of features and a linear discriminant analysis classifier. While this significantly outperforms the LOSO approach which would be the closest equivalent comparison, the MDL approach is where TIDNet with EA and mixup score very well, which is (detectable with $p < 0.004$, but not significant after correction) 5.867% higher than the results by He *et al*. Furthermore, this score is 5.7% higher than the original *subject-specific* SCN [2], while the MDL SCN has a more modest 3.7% improvement (significances are unknown as only mean results were published).

With the P300 dataset we see improvement over baseline performance across all permutations considered. Notably, we can make a comparison to previous work where multiple P300 datasets are leveraged to augment performance [19]. Their performance on this dataset, *without seeing it*, but pre-trained on 5 other P300 tasks is $0.806 \pm 0.004\,3$ (mean and standard deviation across ten fold cross-validation). They then fine-tuned the pre-trained model on 80% of the target *dataset* (this same P300 dataset), notably *randomly from all subjects*. To compare our own performance to theirs, the highest performing results are 0.824 ± 0.068 LMSO and $0.856 \pm 0.061\,7$ MDL (which similarly includes data from *all* subjects, but leverages 83% of the dataset). Our results exhibit higher variability, but appear on par with their results, without any pre-processing beyond our rough down-sampling, and **without pre-training with five additional datasets**.

Considering the ERN dataset, TIDNet in conjunction with EA and/or mixup again outperforms the baseline, in a notably significant fashion, although these are relatively minor increases. This implies that the performance benefit, although mild, is highly consistent on a subject-to-subject basis. Furthermore, while EEGNet (our employed baseline) to our knowledge was the highest previously performing DNN on this dataset, and our results indicate that TIDNet with EA might be better, a previous method using covariance-based classification well exceeded our own performance scoring 0.846 mean AUROC to our 0.748, although this performance notably was the result of 500 bagged models trained, in aggregate, across *all* subjects and included meta-features beyond raw trial data (see https://github.com/alexandrebarachant/bci-challenge-ner-2015 for more information). Considering the closest comparison to our own arrangement, that similarly employed LMSO cross-validation, the performance is much lower at 0.729 AUROC, 0.024 lower than the best performing TIDNet configuration.

4.1. Targetted MDL vs. fine-tuning

Comparing our DG+DA fusion approach to pure DA shows unanimous performance gains across datasets, tasks, and subject types. For the MMI data, fine-tuning TIDNet in fact outperforms prior work under the exact same stratification of subjects and target data [15] by 1.70%, 4.37% and 7.50% across 2, 3, and 4 way classification respectively, again observing that TIDNet performs progressively better than shallow models in the more complex 4-way classification task from these data. Despite this *increase* over prior work, our MDL approach further increases performance by 0.68%, 2.78% and 2.72% in the 2-, 3-, and 4-way tasks respectively. This appears to be the highest performing overall subject-specific performance achieved with this dataset. Notably, taking a 3 s window rather than a 6 s window actually hampers performance generally [15]. Taking a 6 s window, the target subject MMI performance

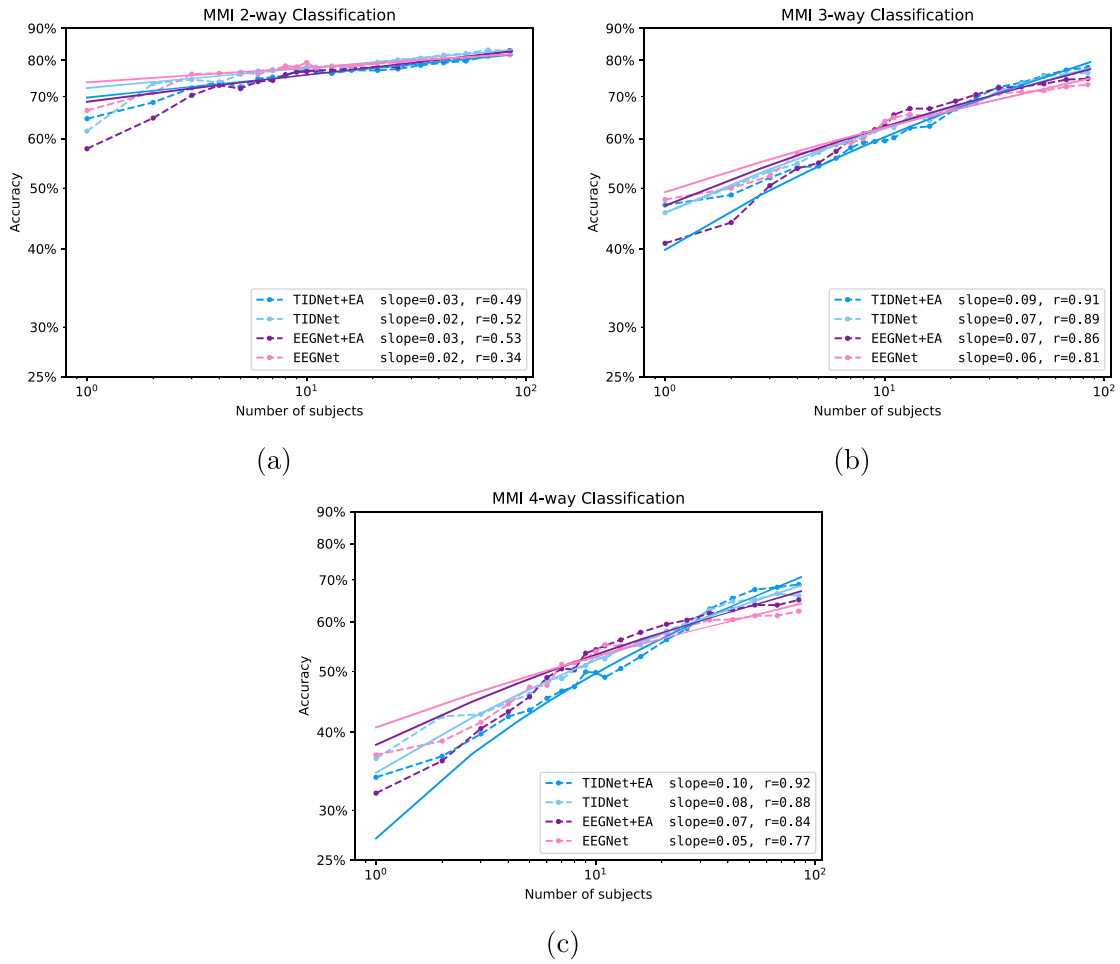

Figure 4. Mean ten-fold cross-validation performance of TIDNet and EEGNet on the 2, 3 and 4-way classification tasks of the MMI dataset plotted against the number of subjects used for training. For each model, the addition of highest performing TI and unaugmented configurations are compared. The dashed lines join the 20 different subsets of training subjects, while the solid lines (with like colour of solid line corresponding to like model/augmentation dashed line) are a log-space regression of each series, modelling the performance trend for each configuration.

using MDL dramatically improves to 92.33%, 90.33% and 84.22%. To the best of our knowledge, these are the highest published results using this dataset *by far* (with an improvement of 5.84%, 11.28% and 15.71% for each task respectively). It is worth noting that with the expanded window in the 2-way task, nearly half of the subjects had their trials classified at rates greater than 95% (50 of 105) and nearly a third overall (31 of 105) achieved 100% accuracy, and the over 90% accuracy in the 3-way task is MI between left hand, right hand and rest.

4.2. Subject regression

By plotting the performance of TIDNet and EEGNet, both with and without DG against the number of subjects used for training, we model some of the differences between these conditions as a function of training subjects. Figure 4 shows the mean performance across 10 different cross-validation stratifications for 20 logarithmically spaced subsets of subjects on a log–log plot. We add log-space regression lines fit using individual test performances (of all points, not of the mean line) against the number of subjects used to train the model. When performing the regression, we exclude subsets with fewer than five subjects, as the mean performance below this point is not well suited to a logarithmic relationship. The 2-way figure stands out due to the relative saturation of performance levels compared to the 3 and 4 way tasks. It gives context to the scale of improvements in performance, as by 20 training subjects each configuration is well within 5% of the maximum value. Other features that stand out include the tendency for the DG-free shallow network to have good performance with fewer subjects, and for this performance to trend below the remaining configurations as more subjects are added. Similarly for TIDNet+EA, performance begins quite low, but makes steady improvement coming out ahead in terms of absolute performance and trend in the 3- and 4-way tasks. The addition of EA does not seem particularly helpful to TIDNet in the two-way task, but given the saturation of the performance, it is unclear. Otherwise, in terms of both trend and peak performance, EA seems consistently beneficial to EEGNet as well as TIDNet.

While each regression line is determined with $p << 10^{-5}$, the r-values are quite low for the 2-way classification (sub-figure (a)) with 0.49, 0.52, 0.53, and 0.34 for the TIDNet+EA, TIDNet, EEGNet+EA, and EEGNet lines, again indicating that, at least with this level of performance, the marginal addition of subjects provides little performance gain, but the shallow model without DG is the most affected. In contrast, the 3- and 4-way task exhibit much higher r-values, with EEGNet alone consistently having the lowest r-values in its group. In terms of model slope, we find a mostly consistent pattern of increasing slope values in the order of EEGNet, EEGNet with EA, TIDNet, and then TIDNet with EA. The only inconsistency was the promotion of EEGNet with EA to have the highest slope in the 2-way task, with all other positions staying consistent. This strongly suggests, at least for this dataset, that there is a greater capacity for leveraging additional subjects when using TIDNet and EA.

5. Discussion

We suggest that the weakest aspect of TIDNet is an unnecessary loss of spatial information caused by treating each channel as an element of a largely unordered list, one that is crucially one-dimensional rather than that which would naturally better represent a 2D or 3D sensor layout. We preserve this for two main reasons: in this way, a deeper architecture is much more computationally tractable and, as of yet, there is no established common way to represent the sensor layout topology for use by a neural network. While there exists two somewhat common methods to do this—direct pixel to channel mapping [31] and an interpolated image of sensor values assigned locations based on azimuthal equidistant projection of sensor locations on the scalp [53] (and similarly with an MEG helmet in our previous work [26])—it is our opinion that neither of these has been established as superior even to the single list.

A notable difference in how data is presented to both TIDNet as well as the reference designs in this work is the normalization of trials between -1 and 1 rather than by z-score normalization or like variants [2, 3]. Early in our experimentation we noticed a slight performance improvement when using our normalization scheme over z-scoring. Given the sensitivities that neural networks have to the exploding and vanishing gradient problem as a result of the distribution of incoming data (whether at the input, or within the network) [34, 54], we suspected that incoming values were poorly distributed, potentially undermining the theory grounding things like weight-initialization schemes that are meant to prevent exploding or vanishing gradients [30, 54]. Our concern was specifically that maximum values where too widely distributed. Figure 5 plots the distribution of absolute maximum trial values after z-score normalization of the IV-2a dataset. Notice that there was quite a skewed distribution, far from the sorts of normally distributed activations that neural networks theory prefers [54] to consider. While these represent all the *absolute maximums*, and thus are not a direct parallel to activations or generally inputs as such, we decided that our strict confinement of these extremes to two values should mitigate this possible concern, while leaving the distribution of internal activations to be addressed by batch normalization [34]. While this calls to question whether z-scoring (along channels) is an appropriate input normalization for raw BCI-trial DNN classifiers, it is far from conclusive. Future work would do well to consider this problem more directly and whether alternative normalization schemes would further improve results.

Above, we ultimately ignored any provided EOG channels rather than use them to perform any ocular signal removal. It is well known that these signals obfuscate the underlying neural signals, and are an undesirable feature of classification of *exclusively neural* signals. While this is known, we were unaware of any previous work that investigated the effects of EOG channels/ocular artifact rejection and shallow DNN variants like EEGNet and SCN in particular. In an effort to characterize how these signals affected TIDNet and our shallow references, we have appended a short ablation study in Appendix B, which found that even when including EOG channels, or when simple ocular artifact rejection (via highly correlated independent component analysis (ICA) components to EOG channels), a similar trend emerged as in the main body of this work, where TIDNet excelled when augmented with EA and mixup. We suggest that this is due to a higher reliance on signals that originate from neural sources but do not confirm this, feeling it was out of the scope of this work. While we feel the *black box* nature of DNNs is overstated when compared to the sometimes spurious collection of features that enable 'classical' classifiers to work, DNNs remain a very obfuscated tool nonetheless. It is therefore hard to say what exactly the DNNs are leveraging to make decisions. Here, we relied on the fact that performance carried over to completely unseen subjects to justify the applicability of TIDNet as a classifier for BCI. We do however believe that the use of 'explainable AI' (XAI) techniques will be crucial using DNN classifiers in BCI moving forward, if only for its potentially unique ability for *hypothesis through data-mining* [55], where even potentially *new* hypotheses might be developed by considering the operations learned by DNNs, rather than simply endeavouring to show that DNNs pick up on what are considered to be known features.

Strangely, a few of the shallow reference configurations, when paired with EA, failed to score above chance. It is hard to say what exactly caused this, but we suggest that a possible explanation might be the inability of the shallow classifier to separate

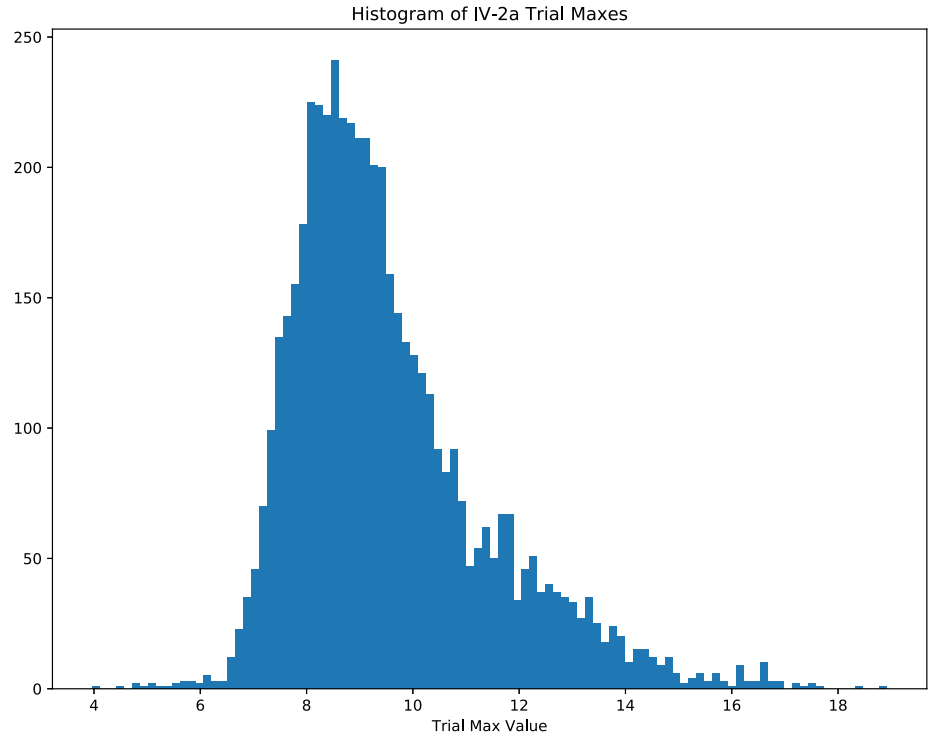

Figure 5. Histogram of absolute maximum trial values for dataset IV-2a. It has a difficult to characterize distribution with a noticeably heavy right tail, which implies at the very least that points are not consistently within a certain range, which has likely detrimental effects on the backpropagated gradients.

trials when they are *better* distributed. Consider that when the distributions are sufficiently disparate, an ensemble of simple internal classifiers (one for each disparate grouping) might discover rules for each cluster. When the clusters are more aligned, the complexity of the arrangement may have been too difficult to overcome without greater degrees of freedom. This may also explain a few other observations: in figure 4, EEGNet+EA is consistently the poorest performing when only a handful of subjects are available, and also why in Appendix B, when ocular/EOG components were increasingly controlled for, the augmented shallow models were more strongly *negatively* affected. This latter point would be consistent with the ocular/EOG signals serving as a high signal-to-noise-ratio feature that could be used to, in some sense, *select for* the appropriate internal model. This is highly speculative to say the least, and even XAI techniques may serve simply to obscure the truth in the matter, but notably, we never noticed a case where TIDNet in conjunction with EA was similarly incapable of scoring above chance.

We originally hypothesized that combining EA and mixup would imbue performance increases akin to added data scale in a naturally complimentary way. Our empirical results are clear that using both *consistently* increases performance but, in terms of magnitude of increase, EA alone is many times better performing. Currently, using mixup alone without first aligning the data is never particularly helpful, which is consistent with the idea that many trials are proximate *after* alignment, but are not sufficiently nearby *before* alignment to take advantage of a vicinal prior. This was true for our baseline models as well, and we would therefore recommend their use in other LO/MSO situations, whether TIDNet is used or not. Furthermore, it should be noted that we narrow our focus of potential DG approaches to *domain invariant features* and *data augmentation* (EA and *mixup*, respectively), but there are other possible avenues of DG, MDL, and TL that have developed in recent years from the DNN research community [38, 56, 57].

We elected to compute possibly multiple alignments for each subject by performing EA with respect to each recorded file (no file had more than one person's data). This had the potential benefit of also addressing variations between sessions (in these datasets files and sessions are synonymous), which are an additional source of domain shift, but a large number of sessions, with less data to leverage per session may contribute to a poorer $\bar{R}$. In this vein we note that the relatively poorer performances of the N2PC and P300 datasets correlated with the extremes of this process: a single alignment procedure and up to 20 separate alignments respectively. It may be necessary to consider this trade-off in future work, but further study is required.

These results show a consistent performance gain using DG/MDL, most consistently prominent with the deeper TIDNet model, for offline public datasets—a step that is crucial for establishing a comparable benchmark to other work. Perhaps the most

important next step is determining how exactly these techniques are applied to practical settings. It should be kept in mind that even awareness of established performance benchmarks serves as a sort of data *over-fitting*, and performance is likely exaggerated over novel settings. That said, LOSO/LMSO models (without EA) have a clear path to practical classification of new subjects with data collected from similar hardware and settings: offline global models serve as ready-to-use classifiers for new subjects. Slightly more challenging is how MDL could be leveraged in practice. The simplest answer seems to be to develop continually adaptive classifiers that would not otherwise be practical by fine-tuning. An 'online' paradigm where a single model could be constantly optimized by re-training with additional incoming data from new subjects. Given our results from section 4.1, this is preferable to fine-tuning as the target performance is both higher and less (if at all) destructive towards *non-target* subject accuracy. A single model then remains nearly universal, despite the addition of small sets of novel subject data (that is appropriately over-sampled as we do above). For example, starting from an original model M_0 trained with the original data pool D_0, a new *target* session is recorded: D_1. During this session, M_0 can be used for an already functional BCI, but during or prior to subsequent sessions for data D_{n+1}, new models M_n are trained using $\bigcup_{i=0}^{n} D_i$. This discussion however is yet to consider EA, which presents a slight challenge to this approach, particularly for the first session, as it requires a mean covariance matrix of trials. It relies on at least some representative domain data to align subsequent data, and so far these are notably *trials*, not simply resting data (although it is possible this may work as well). While we did not model the number of trials needed to successfully reap the benefits of EA, we explored the worst-case scenario of classifying *completely unaligned* trials (that is, raw trials) with a model previously trained with aligned data. Looking at non-target MDL performance (the same procedures as 4.1) for the 3- and 4-way MMI tasks (2-way performance was maximized without EA), and the ERN task, we find mean (completely unseen subject) performance drops to 69.07%, 56.14%, and 0.649 AUROC respectively. Interestingly, the MMI performance drops to very similar levels as successful prior work; *i.e.* Dose *et al* achieved LMSO accuracies of 69.82% and 58.58% respectively [15]. On the other hand, the ERN performance fared worse, scoring 0.035 lower than a non-augmented EEGNet, but these decreases in performance were much less dramatic than we expected, and indicate that EA could likely still be used in an *online-bootstrapping* fashion as we describe.

Another natural extension of this work is to use similar DG/MDL techniques to improve performance for similar tasks recorded in different recording contexts, e.g. different recording equipment with their own number of channels, sampling frequencies, and analog-to-digital conversions. While prior work has learned like-tasks embeddings, creating a projection to a common task subspace for later classification [19], integrating more explicit DG methodology or a MDL approach may further improve their performance. We hope to develop this idea further and invite others to consider integrating these approaches.

The large improvement in performance we observed when expanding our trial window to 6 s in section 4.1, and the relatively little improvement over prior work we saw with the N2PC dataset (eight channels) would make it unsurprising if TIDNet was not well suited to data collected with a limited number of electrodes. In general, it appeared to us that our approach is best used with *long* sequences and increased electrode densities and when no particular *bands* of import have been selected for. Part of our goal is to develop classifiers that are successful *without* the need for dataset-specific pre-processing, and greater performance using longer sequences and many electrodes suits this nicely.

Our results should help inform how TL in BCI, and particularly with DNNs, is performed. We showed that disambiguating DG and domain adaptation has particular merit, and that leveraging them in a more complementary fashion can be beneficial. TIDNet's performance was improved through both fine-tuning (DA), and also generally through a mixture of EA + mixup (DG), but the largest improvements are seen through our proposed MDL framework that leverages both DG *and* DA, where a single model is well suited to new (*i.e.* target) subjects with little training data *and* to completely unseen ones. TIDNet, our proposed deeper DNN, while not necessarily the strongest choice *before* DG, is most consistently preferable after, which suggests to us that deeper models (TIDNet or otherwise) have thus far been prohibited by the poor empirical estimation of $P(x, y)$, and that deeper models may excel both with limited data but appropriate application of DG, and with the continued addition of more subjects (and consequently data). In either case, this results in a single, more universal classifier than prior convention. Our results are all empirically validated using publicly accessible data, and the code used is available in its entirety, which we hope will help continue this line of research by others.

Acknowledgment

We would like to emphasize our thanks to those who make data they have recorded publicly accessible to all. Rudzicz is supported by a CIFAR Chair in Artificial Intelligence. Also, this work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [20ZS1100, Core Technology Research for Self-Improving Integrated Artificial Intelligence System].

Table A1. Listing of the hyperparameters that differed between datasets. Starting point in most cases was 20 epochs, batch size of 16, 3 epochs of exponential weight smoothing (EWMA weights) and 8.0 mixup. This remained the most common condition. ERN dataset required the only divergence between TIDNet and reference to maintain both performances high.

Dataset		IV-2a	MMI	P300	ERN	N2PC
TIDNet	Epochs	100	30	20	20	20
	Batch Size	60	16	16	4	16
	EWMA weights	5	5	3	15	3
	Mixup	2.0	8.0	8.0	8.0	8.0
Reference (SCN or EEGNet)	Epochs	100	30	20	100	20
	Batch Size	60	16	16	32	16
	EWMA weights	5	5	3	15	3
	Mixup	2.0	8.0	8.0	8.0	8.0

Appendix A. Hyperparameters

A.1. TIDNet

TIDNet is modifiable by several first-order hyperparameters that have the largest impact on its structure (aside from what has already been presented in section 3.1). These are, and under this work have the following values:

- Number of temporal layers: 2
- Number of temporal filters: 32
- Max temporal pooling (width and stride): 15
- Number of DenseFilters: 2
- Bottleneck factor within DenseFilter: 3
- DenseNet-style growth factor (added filters per DenseFilter): 24

A.2. Dataset-specific

As mentioned in section 3.6, there was minimal variation in hyperparameters between datasets, the TIDNet architecture remained fixed, as did parameters like label smoothing, optimizer, weight decay and more. The only points of variation in hyperparameters are listed in table A1. The optimization procedure was entirely heuristic, beginning with a learning rate of 0.001, batch size of 16, 20 epochs and exponential weight smoothing for the final 3 epochs. Notice that most values employed remained mostly the same. We would recommend beginning with similar hyperparameters if trying with new datasets, but expect there are potential performance gains to be made by optimizing these, and the more specific parameters of TIDNet, e.g. the number of layers of each stage, or the growth rates of the densenet-like spatial stage.

Appendix B. Effect of EOG electrodes

While it is well known that EOG electrodes may increase performance (and *will* confound results to say the least), we were unaware of any previous efforts to determine the susceptibility of common DNN architectures (such as EEGNet or SCN) in particular to EOG signals. We performed a short ablation study that compared the effects of: 1. Simply not using EOG electrodes (the results presented in the main body of this work), 2. Including the EOG electrodes during training, and 3. Performing a simple Pearson-correlation-based ICA component artifact rejection on the three datasets that included EOG channel recordings: IV-2a, P300 and ERN. In particular, the ICA rejection involved decomposing the N_c EEG channels into $\lfloor \frac{N_c}{2} \rfloor$ components using *fastica* [58]. Then using the automated (simply thresholded Pearson correlation values to EOG channels) EOG rejection provided by the MNE library (https://mne.tools/stable/generated/mne.preprocessing.ICA.html?mne.preprocessing.ICA.find_bads_eog#mne.preprocessing.ICA.find_bads_eog), no more than 3 (most correlated) components are rejected before translating ICA components back into EEG channel space. Table B2 shows the differences in performance across each of these three configurations across all three datasets. While the sheer *number* of comparisons make statistical analysis of all these performances difficult, we elected to include these as an appendix to discuss the trends we believed these data indicated.

It is immediately clear from the first two blocked rows of table B2 that the EOG electrodes and signals increased classification accuracy with the IV-2a dataset. With a continual decrease in performance with each measure introduced to remove EOG. In fact, performing classification using *only* the three EOG electrodes produced an average classification accuracy of 59.51% and 61.09% accuracy for TIDNet and SCN respectively (paired Wilcoxon test $p < 0.14$). The SCN seemed to perform better with no training augmentations, but TIDNet consistently made good use of the training augmentations, particularly so even when EOG correlated components were extracted (ICA-EOG rows), in which case SCN was more catastrophically affected.

The ERN dataset on the other hand indicates that EOG electrodes are relatively detrimental with the *+EOG* condition being the worse performing in nearly all cases, although interestingly, this effect is somewhat mitigated with *EA* and *EA+mixup (Both)*

Table B2. Results of short ablation study with respect to EOG/ocular signals, where the three datasets with explicitly recorded EOG channels are included (+EOG), not included (−EOG), or used to reject correlated ICA components (ICA-EOG). Performance for all augmentations considered for datasets IV-2a (accuracy), P300 and ERN (AUROC * 100), with and without MDL.

		TIDNet				Reference			
Dataset	Signal	Base	Mixup	EA	Both	Base	Mixup	EA	Both
IV-2a	+EOG	70.5	68.7	72.1	73.4	71.8	68.7	71.7	70.5
	−EOG	58.3	57.0	65.4	65.2	59.8	60.6	64.3	64.6
	ICA-EOG	55.5	56.1	63.5	64.9	59.1	55.2	59.2	55.5
IV-2a+MDL	+EOG	77.7	75.9	83.8	84.2	77.6	74.8	80.0	76.0
	−EOG	66.7	69.8	79.0	79.4	72.0	69.1	77.7	77.4
	ICA-EOG	62.2	66.1	75.6	76.0	67.2	64.2	66.3	62.3
P300	+EOG	83.2	82.6	83.0	82.8	81.9	81.8	82.1	81.7
	−EOG	82.3	82.3	82.3	82.4	82.0	81.6	81.9	81.7
	ICA-EOG	81.9	81.8	81.9	82.0	81.8	81.6	81.6	81.4
P300+MDL	+EOG	86.0	86.2	86.4	86.0	85.6	85.3	85.6	85.3
	−EOG	85.6	86.0	85.6	85.6	85.7	85.5	85.7	86.4
	ICA-EOG	84.5	85.1	84.8	85.1	84.6	84.6	84.6	84.4
ERN	+EOG	53.5	54.9	74.8	71.7	70.7	69.6	72.1	72.0
	−EOG	66.4	64.2	74.4	74.8	70.2	71.2	72.4	72.4
	ICA-EOG	61.7	63.4	72.1	72.1	63.8	63.2	67.9	67.9
ERN+MDL	+EOG	69.6	73.2	76.4	74.8	68.4	65.5	73.8	73.8
	−EOG	77.8	75.2	81.6	81.7	81.0	79.9	79.7	79.4
	ICA-EOG	79.0	63.4	80.6	80.1	80.4	75.0	74.6	72.5

conditions. Otherwise however, we again see the reference model not benefiting as much as TIDNet from the *EA* and *EA+mixup (Both)* conditions, with the best performing condition for EEGNet being the *Base* condition. Notably however, it performs very similarly to TIDNet *with* augmentations. It should also be noted, that unlike IV-2a, single digits here are a 0.01 AUROC, which is probably less of a material difference than a percentage point.

The P300 dataset shows a similar decreasing pattern of performance, but is markedly less pronounced, with the decreases for the most part uniform between the two architectures. It is also worth noting that we considered the effect of simply ignoring the *A1* and *A2* ear reference channels recorded for the P300 dataset (at the same time, simply ignoring the EOG electrodes). We do not add them here, but interestingly its effect was largely the same as the *ICA-EOG* data configuration. Again, much like the other conditions with the P300 dataset, the effects were small and relatively uniform between models and training augmentations.

Our observations from these extended tests further compound the observations in the main body, that TIDNet excels when data is aligned, making better use of more similarly distributed data. On the whole, the reference implementations tended to respond more poorly to training augmentations the more ocular effects where removed, juxtaposed with the fact that in 16 of the 18 data configurations in table B2, the augmented TIDNet outperformed the base reference. The two instances where this was not the case, were neither statistically detectable differences, nor relatively large, with absolute differences of 0.2 and 0.3 expressed in the metrics above. We therefore suggest that the shallower EEGNet and SCN models might be more susceptible to strong confounding signals like ocular artifacts, and augmented TIDNet training is the safer choice in general.

ORCID iD

Demetres Kostas https://orcid.org/0000-0001-9516-8145

References

[1] Lotte F, Bougrain L, Cichocki A, Clerc M, Congedo M, Rakotomamonjy A and Yger F 2018 A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update *J. Neural Eng.* **15** 031005

[2] Schirrmeister R T, Springenberg J T, Dominique L, Fiederer J,, Glasstetter M, Eggensperger K, Tangermann M, Hutter F, Burgard W and Ball T 2017 Deep learning with convolutional neural networks for EEG decoding and visualization. *Hum. Brain Mapp.* **38** 5391-420

[3] Lawhern V J, Solon A J, Waytowich N R, Gordon S M, Hung C P and Lance B J 2018 EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces *J. Neural Eng.* **15** aace8c

[4] Roy Y, Banville H, Albuquerque I, Gramfort A, Falk T H and Faubert J 2019 Deep learning-based electroencephalography analysis: a systematic review *J. Neural Eng.* **16** 051001

[5] Craik A, He Y and Contreras-Vidal J L 2018 Deep learning for electroencephalogram (EEG) classification tasks: A review *J. Neural Eng.* **16** 031001

[6] Ozdenizci O, Wang Y, Koike-Akino T and Erdogmus D 2019 Transfer learning in brain-computer interfaces with adversarial variational autoencoders *2019 9th Int. IEEE/ Conf. on Neural Engineering (NER)* IEEE pp 207–10

[7] Hartmann K G, Schirrmeister R T and Ball T 2018 Eeg-gan: Generative adversarial networks for electroencephalograhic (EEG) brain signals arXiv:1806.01875

[8] Devlin J, Chang M-W, Lee K and Toutanova K 2018 BERT: pre-training of deep bidirectional transformers for language understanding *CoRR* arXiv:1810.04805

[9] Blankertz B, Kawanabe M, Tomioka R, Hohlefeld F, Müller K-R and Nikulin V V 2008 Invariant common spatial patterns: alleviating nonstationarities in brain-computer

interfacing Platt J C, Koller D, Singer Y and Roweis S T eds (Red Hook, NY: Curran Associates, Inc.) *Adv. Neural Inf. Proc. Syst. 20* pp 113–20
[10] He H and Wu D 2019 Transfer learning for brain-computer interfaces: A euclidean space data alignment approach *IEEE Trans. Biomed. Eng.* **67** 1–1
[11] Zanini P, Congedo M, Jutten C, Said S and Berthoumieu Y 2018 Transfer learning: A riemannian geometry framework with applications to brain-computer interfaces *IEEE Trans. Biomed. Eng.* **65** 1107–16
[12] Zheng W L and Lu B L 2016 Personalizing EEG-based affective models with transfer learning *Proc. 25th Int. Joint Conf .on Artificial Intelligence* AAAI Press New York 2016 2732–8
[13] Fahimi F, Zhang Z, Goh W B, Lee T-S, Ang K K and Guan C 2019 Inter-subject transfer learning with an end-to-end deep convolutional neural network for EEG-based BCI *J. Neural Eng.* **16** 026007
[14] Schwemmer M A, Skomrock N D, Sederberg P B, Ting J E, Sharma G, Bockbrader M A and Friedenberg D A 2018 Meeting brain–computer interface user performance expectations using a deep neural network decoding framework *Nat. Med.* **24** 1669–76
[15] Dose H, Møller J S, Iversen H K and Puthusserypady S 2018 An end-to-end deep learning approach to MI-EEG signal classification for BCIs *Expert Syst. Appl.* **114** 532–42
[16] Völker M, Schirrmeister R T, Fiederer L D J, Burgard W and Ball T 2018 Deep transfer learning for error decoding from non-invasive EEG pp 1–6 arXiv:1710.09139
[17] Lin Y-P and Jung T-P 2017 Improving EEG-based emotion classification using conditional transfer learning *Front. Hum. Neurosci.* **11** 1–11
[18] Xu G, Shen X, Chen S, Zong Y, Zhang C, Yue H, Liu M, Chen F and Che W 2019 A deep transfer convolutional neural network framework for EEG signal classification *IEEE Access* **7** 112767–76
[19] Ditthapron A, Banluesombatkul N, Ketrat S, Chuangsuwanich E and Wilaiprasitporn T 2019 Universal Joint Feature Extraction for P300 EEG Classification Using Multi-Task Autoencoder *IEEE Access* **7** 68415–28
[20] Jolly B L K, Aggrawal P, Nath S S, Gupta V, Grover M S and Shah R R 2019 Universal EEG encoder for learning diverse intelligent tasks *Proc. - 2019 IEEE 5th Int. Conf. on Multimedia Big Data, BigMM 2019* pp 213–18
[21] Banville H, Moffat G, Albuquerque I, Engemann D A, Hyvarinen A and Gramfort A 2019 Self-supervised representation learning from electroencephalography signals *IEEE Int. Workshop on Mach. Learn. Sig. Proc.*
[22] Esteva A, Kuprel B, Novoa R A, Ko J, Swetter S M, Blau H M and Thrun S 2017 Dermatologist-level classification of skin cancer with deep neural networks *Nature* **542** 115–18
[23] Wang Y, Yao Q, Kwok J and Ni L M 2019 Generalizing from a few examples: A survey on few-shot learning arXiv:1904.05046
[24] Zhang H, Cisse M, Dauphin Y N and Lopez-Paz D 2017 mixup: Beyond empirical risk minimization *CoRR* arXiv:1710.09412
[25] Zhu X, Li P, Li C, Yao D, Zhang R and Xu P 2019 Separated channel convolutional neural network to realize the training free motor imagery BCI systems *Biomed. Signal Process. Control* **49** 396–403
[26] Kostas D, Pang E W and Rudzicz F 2019 Machine learning for MEG during speech tasks *Sci. Rep.* **9** 1609
[27] Xu B, Wang N, Chen T and Li M 2015 Empirical evaluation of rectified activations in convolutional network *CoRR* arXiv:1505.00853
[28] Bai S, Kolter J Z and Koltun V 2018 An empirical evaluation of generic convolutional and recurrent networks for sequence modeling *CoRR* arXiv:1803.01271
[29] Gemein L A W, Schirrmeister R T, Chrab Aszcz P, Wilson D, Boedecker J, Schulze-Bonhage A, Hutter F and Ball T 2020 Machine-learning-based diagnostics of EEG pathology *Tech. Rep.* arXiv:2002.05115
[30] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* IEEE 1–17
[31] Joshi R, Goel P, Sur M and Murthy H A 2018 Single trial P300 classification using convolutional LSTM and deep learning ensembles method *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol 11278 LNCS pp 3–15
[32] Croce P, Zappasodi F, Marzetti L, Merla A, Pizzela V and Chiarelli A M 2018 Deep Convolutional Neural Networks for feature-less automatic classification of independent components in multi-channel electrophysiological brain recordings *IEEE Trans. Biomed. Eng.* **66** 9294
[33] Huang G, Liu Z, Van Der Maaten L and Weinberger K Q 2018 Densely connected convolutional networks *Tech. Rep.* arXiv:1608.06993
[34] Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift *CoRR* abs/1502.0
[35] Yger F, Berar M and Lotte F 2017 Riemannian approaches in brain-computer interfaces: A review *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **25** 1753–62
[36] Bronstein M, Bruna J, Szlam A, Bresson X and Lecun Y 2017 Geometric deep learning on graphs and manifolds (tutorial) *NIPS 2017* pp 18–42
[37] Farquhar J and Hill N J 2013 Interactions between pre-processing and classification methods for event-related-potential classification: best-practice guidelines for brain-computer interfacing *Neuroinformatics* **11** 175–92
[38] Li D, Zhang J, Yang Y, Liu C, Song Y-Z and Hospedales T M 2019 Episodic training for domain generalization CoRR arXiv:1902.00113
[39] Goldberger A L *et al* 2000 PhysioBank, PhysioToolkit and PhysioNet: components of a new research resource for complex physiologic signals *Circulation* **101** e215–e220
[40] Tangermann M *et al* 2012 Review of the BCI competition IV *Front. Neurosci.* **6** 1–31
[41] Sakhavi S, Guan C and Yan S 2018 Learning temporal information for brain-computer interface using convolutional neural networks *IEEE Trans. Neural Netw. Learn. Syst.* **29** 5619–29
[42] Schalk G, Mcfarland D J, Hinterberger T, Birbaumer N and Wolpaw J R 2004 BCI2000: A general-purpose brain-computer interface (BCI) System *IEEE Trans. Biomed. Eng.* **51** 1034–43
[43] Citi L, Poli R and Cinel C 2010 Documenting, modelling and exploiting P300 amplitude changes due to variable target delays in Donchin's speller *J. Neural Eng.* **7** 056006
[44] Stivers J, Mousavi M and De Sa V 2017 Deep recurrent convolutional neural networks for classifying P300 BCI signals *7th Graz Brain-Computer Interface Conference 2017*
[45] Solon A J, Lawhern V J, Touryan J, McDaniel J R, Ries A J and Gordon S M 2019 Decoding P300 variability using convolutional neural networks *Front. Hum. Neurosci.* **13** 1–12
[46] Shan H, Liu Y and Stefanov T 2018 A simple convolutional neural network for accurate P300 detection and character spelling in brain computer interface *IJCAI Int. Conf. on Artificial Intelligence* vol 2018-July pp 1604–10
[47] Cecotti H and Gräser A 2011 Convolutional neural networks for P300 detection with application to brain-computer interfaces *IEEE Trans. Pattern Anal. Mach. Intell.* **33** 433–45
[48] Matran-Fernandez A and Poli R 2017 Towards the automated localisation of targets in rapid image-sifting by collaborative braincomputer interfaces *PLoS One* **12** 1–28
[49] Reddi S J, Kale S and Kumar S 2018 On the convergence of adam and beyond *ICLR* pp 1–23

[50] He T, Zhang Z, Zhang H, Zhang Z, Xie J and Li M 2018 Bag of tricks for image classification with convolutional neural networks *CoRR* arXiv:1812.01187
[51] Tan C, Sun F, Kong T, Zhang W, Yang C and Liu C 2018 A survey on deep transfer learning *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol 11141 LNCS pp 270–9
[52] Kemker R, McClure M, Abitino A, Hayes T L and Kanan C 2018 Measuring catastrophic forgetting in neural networks *32nd Conf. on Artificial Intelligence, AAAI 2018* pp 3390–8
[53] Bashivan P, Rish I, Yeasin M and Codella N 2016 Learning representations from eeg with deep recurrent-convolutional neural networks *ICLR 2016* pp 1–15
[54] Glorot X and Bengio Y 2010 Understanding the difficulty of training deep feedforward neural networks *J. Mach. Learn. Res.* **9** 249–56
[55] Cichy R M and Kaiser D 2019 Deep neural networks as scientific models *Trends Cogn. Sci.* **23** 305–17
[56] Carlucci F M, D'Innocente A, Bucci S, Caputo B and Tommasi T 2019 Domain generalization by solving jigsaw puzzles *CoRR* arXiv:1903.06864
[57] Shankar S, Piratla V, Chakrabarti S, Chaudhuri S, Jyothi P and Sarawagi S 2018 Generalizing across domains via cross-gradient training *CoRR* arXiv:1804.10745
[58] Hyvärinen A and Oja E 2000 Independent component analysis: algorithms and applications *Neural Netw.* **13** 411–30