Challenges for cognitive decoding using deep learning methods

Armin W. Thomas *,♢, Christopher Ré ■, and Russell A. Poldrack ★,♢

★ Stanford Data Science, Stanford University, Stanford, CA, USA
♢ Department of Psychology, Stanford University, Stanford, CA, USA
■ Department of Computer Science, Stanford University, Stanford, CA, USA

*Correspondence: athms@stanford.edu (A. W. Thomas)

Keywords: cognitive decoding, deep learning, neuroimaging, explainable artificial intelligence, transfer learning, robustness

In cognitive decoding, researchers aim to characterize a brain region’s representations by identifying the cognitive states (e.g., accepting/rejecting a gamble) that can be identified from the region’s activity. Deep learning (DL) methods are highly promising for cognitive decoding, with their unmatched ability to learn versatile representations of complex data. Yet, their widespread application in cognitive decoding is hindered by their general lack of interpretability as well as difficulties in applying them to small datasets and in ensuring their reproducibility and robustness.

We propose to approach these challenges by leveraging recent advances in explainable artificial intelligence and transfer learning, while also providing specific recommendations on how to improve the reproducibility and robustness of DL modeling results.
The promise of deep learning

Over the last decade, deep learning (DL; see Glossary and [1]) methods have revolutionized many areas of research and industry with their ability to learn highly versatile representations (or concepts) from complex data. A defining feature of DL methods is that they sequentially apply stacks of many simple, but nonlinear, transforms to their input data, allowing them to gain an increasingly abstracted view of the data. At each level of the transform, new representations of the data are built by the use of representations from preceding layers. The resulting high-level view of the data enables DL methods to capture complex nonlinearities, associate a target signal with highly variable patterns in the data (e.g., when identifying objects in images or transcribing audio recordings), and effectively filter out aspects of the data that are irrelevant to the learning task at hand. A key driver for the empirical success of DL methods is that they are able to autonomously learn these different levels of abstraction from sufficiently large datasets, without the need for extensive data preprocessing or a prior understanding of the mapping between input data and target signal.
This empirical success has recently sparked interest in the application of DL methods to the field of neuroimaging, focused on cognitive decoding [2]. Here, researchers aim to understand the mapping between a set of cognitive states (e.g., answering questions about a math problem vs. a prose story) and the underlying brain activity by training some predictive model to identify these states from measured brain activity [3]. At first sight, DL methods seem ideally suited for these types of analyses, as the mapping between cognitive states and brain activity is often a priori unknown, can be highly variable within [4] and between individuals [5], and is subject to spatial and temporal non-linearities [6].

Yet, the application of DL methods to cognitive decoding analyses also raises several new challenges for researchers who are interested in combining methods from both fields, namely, a general lack of interpretability of DL methods, their overall demand for large training datasets, and difficulties in ensuring the reproducibility and robustness of DL modeling results. Here, we outline each of these challenges and present a set of solutions based on related empirical work and recent methodological advances in both functional neuroimaging and machine learning research.

Opening up the black box

A key challenge for the application of DL models to functional neuroimaging data is the black-box characteristic of DL models, whose highly non-linear nature deeply obscures the relationships between input data and decoding decisions. Thus, even if a DL model accurately decodes a set of cognitive states from functional neuroimaging data, it is not clear which particular features of the data (or combinations thereof) support this decoding. To approach this challenge, functional neuroimaging researchers have begun turning towards research on explainable artificial intelligence (XAI; [7-9]), where techniques are being developed that aim to make the behavior of DL models (and other intelligent artificial agents) understandable for human observers.

One line of research within this field seeks to explain the predictions of DL models by relating these predictions to the features of the input data, thus, making the model interpretable for human observers [10]. On a high level, these explanation techniques can be categorized into those that aim to provide either a global or local explanation of the model’s learned mapping between input data and predictions [8,9]. Global explanations [11,12] are not tied to a specific data sample and seek to provide insights into the characteristics of the data at large that influence the model’s
predictions (e.g., by synthesizing the image for which the model is most certain that it depicts a dog). Local explanations [12-29], on the other hand, are specific for a data sample and seek to provide insights into the mapping between the features of the sample and the resulting prediction of the model (e.g., by identifying those pixels of an image that the model views as evidence for the presence of a dog). Further, explanation techniques can be categorized into those that can be applied post-hoc to the predictions of an already trained model [11-26] and those that are integrated into the training procedure, such that the model is self-explaining and designed a priori to provide human-understandable insights into its learned mappings between data and predictions [27-31].

Here, we focus on local explanation approaches. We provide an overview of a set of representative approaches to this type of XAI in Box 1. Of these approaches, sensitivity analysis [12,22], backward decomposition [13-19,21,24], and attention mechanisms [27-29] are currently most prominent in the functional neuroimaging literature on cognitive decoding with DL models [32-40]. Sensitivity analysis seeks to explain the predictions of a DL model by identifying those features of an input sample to which the model’s prediction responds most sensitively. Backward decompositions, on the other hand, seek to explain individual predictions by sequentially decomposing these predictions in a backward pass into the contributions of the lower-layer model units to these predictions, until the input space is reached and a contribution for each input feature can be defined. Attention mechanisms, in contrast, are inspired by research in neuroscience on perceptual attention [41] and aim to improve the predictive performance of a DL model by focusing its computations on specific aspects of an input sample (through a set of attention weights). Attention mechanisms are thus an integrated feature of the training procedure and cannot be applied post-hoc to the predictions of trained models.

At first sight, the explanations of these three XAI techniques can be difficult for human observers to distinguish, making it difficult to compare the quality of their explanations. To approach this challenge, researchers in machine learning have started developing methods to quantify the quality of these types of explanations [8,42-47]. One prominent approach is to test the faithfulness of an explanation [8,42,44,46]. An explanation can generally be viewed as being faithful [48] if it accurately captures the model’s decision process and thereby identifies those features of the input that are most relevant for the model’s prediction. Accordingly, removing these features from the input (e.g., in an occlusion analysis; [8,42,44]) should lead to a meaningful decline of the model’s predictive performance.
**Box 1. Exemplary approaches to local explainable artificial intelligence.**

We assume that the analyzed model represents some function \( f(\cdot) \), mapping an input \( x \in \mathbb{R}^D \) to some output \( f(x) : \mathbb{R} \to \mathbb{R} \). The presented local explanation approaches \( \eta(\cdot) \) seek to provide insights into this mapping by quantifying the relevance \( r_d \) of each input feature \( d \in D \) for \( f(x) : \eta(\cdot) : \mathbb{R} \to \mathbb{R}^D \) (Fig. 1).

**Occlusion analysis** [20,24,25]: The occlusion analysis is a perturbation analysis, which identifies \( r_d \) by occluding \( x_d \) in the data and measuring the resulting effect on \( f(x) \): \( r_d = \Delta f(x) = f(x) - f(x \times o_d) \). Here, \( o_d \) indicates an occlusion vector (e.g., \( o_d \in [0,1]^D \)), and \( \times \) the element-wise product.

**Interpretable local surrogate model** [26]: A local surrogate model is an interpretable model that is used to explain black-box model predictions by training it to approximate these predictions. In the LIME algorithm [26], \( r \) is quantified by approximating \( f(x) \) for a specific \( x \) with an interpretable model \( g(\cdot) \), e.g., a linear model, where \( g(x) = w^T x \), and which is trained by the use of a set of perturbed versions \( Z \) of \( x \) (e.g., through occlusion): \( \min_w \sum_{z \in Z} \pi_x(z) (f(z) - g(z))^2 \). Here, \( \pi_x \) represents some similarity function weighting each \( z \in Z \) by its similarity to \( x \), while \( r_d \) is given by the corresponding linear model weight \( w_d: r_d = w_d \).

**Sensitivity analysis** [12,22]: Sensitivity analysis defines \( r \) as the locally evaluated partial derivative of \( f(x) \): \( r = f'(x) = \left( \frac{\partial f(x)}{\partial x} \right)^2 \). Accordingly, relevance is assigned to those features \( x_d \) to which \( f(x) \) responds most sensitively.

**Backward decomposition** [13-19,21,24]: Backward decompositions make specific use of the graph structure of DL models by sequentially decomposing \( f(x) \) in a backward pass through the model until the input space is reached. A prominent example is the layer-wise relevance propagation (LRP; [13]) technique: Let \( i \) and \( j \) be the indices of two model units in two successive layers, \( r_j \) the relevance of unit \( j \) for \( f(x) \), and \( r_{i\rightarrow j} \) the part of \( r_j \) that is redistributed from \( j \) to \( i \) during the backward decomposition. To redistribute relevance between layers, several rules have been proposed [49,50], which generally follow from: \( r_i = \sum_j \frac{a_{ij}}{a_{i\rightarrow j} \sum_i a_{i\rightarrow j}} r_j \), where \( a \) and \( w \) represent the input and weights of unit \( i \).

Importantly, LRP assumes that relevance is conserved between layers, such that \( \sum_i r_{i\rightarrow j} = \eta \), \( \eta = \sum_j r_{i\rightarrow j} \), and \( \sum_d r_d = \ldots = \sum_i r_i = \sum_j r_j = f(x) \).

**Attention mechanism** [27-29]: In contrast to the other presented explanation approaches, attention mechanisms are built into and trained with a DL model. Their goal is to help the model focus its computations on specific aspects of \( x \) by the use of attention glimpses \( y \in \mathbb{R}^D \). These glimpses can be implemented by the use of a separate model, which predicts an attention weight \( \alpha_d \) for each \( x_d \) [27,28]. These weights are multiplied with the input (\( y = \alpha \times x \)) and the resulting glimpse is forwarded to the remaining parts of the model (instead of \( x \)). Attention weights are often also interpreted as an estimate of relevance, such that \( r_d = \alpha_d \) [32,42,43].
By the use of this test, researchers in computer vision (CV; [44]) and natural language processing (NLP; [42,43]) have compared the fidelity of explanations resulting from sensitivity analysis and backward decomposition. This work has shown that backward decompositions generally perform better at identifying those features of the input that are most relevant for model predictions. Intuitively, this makes sense, as backward decompositions seek to directly quantify the contribution of each input feature to a specific model prediction. Sensitivity analysis, in contrast, does not evaluate the prediction itself but its local slope, thus identifying features that make the model more or less certain of its prediction, regardless of their actual contribution to the prediction. Similarly, related empirical work [42,43,46] has demonstrated that standard attention mechanisms do not provide faithful explanations of the predictions of DL models, as they fail to identify features of the input that are relevant for the prediction. Attention mechanisms are designed to improve the predictive performance of a DL model by enhancing certain aspects of the input while fading out others. Their decisions on how to weight each input feature are thus tailored to the subsequent computations of the DL model and not designed to identify the relevance of each feature for the prediction.

Figure I. Exemplary approaches to local explainable artificial intelligence.
While functional neuroimaging researchers have also used the occlusion analysis to analyze cognitive decoding models (in “virtual lesion analyses”; [35,51]), these applications have been mostly limited to linear models and to testing whether specific brain regions, which received large weights in a linear model, are actually necessary for an accurate decoding. For functional neuroimaging data, occlusion analyses generally require a clear prior hypothesis on which features (or brain regions) of the input will be tested (e.g., resulting from other research or other explanation analyses), as randomly dropping out individual feature values will otherwise not account for the strong spatial correlation structure inherent to these data.

Taken together, we thus generally recommend the application of backward decompositions to analyze cognitive decoding decisions of DL models (see Box 2 for specific recommendations).

In addition to selecting an XAI technique, functional neuroimaging researchers interested in understanding the decoding decisions of deep cognitive decoding models are also faced with the task of analyzing the resulting dataset of explanations (which is generally equal in size to the original input data, as one explanation is obtained for each input sample). Many current empirical applications of XAI in cognitive decoding simply average individual explanations or analyze them with standard linear models [32-34,36-38]. These types of linear analyses, however, are limited in their ability to identify any non-linear mappings between brain activity and cognitive states that a DL model might have learned. This problem has been similarly encountered in machine learning research. Here, researchers have instead advocated for the application of automated, non-linear, and data-driven approaches to identifying the learned decision behaviors of DL models from explanation data (e.g., through t-distributed stochastic neighbor embedding; [52]). In functional neuroimaging, current methods from network neuroscience [53] also seem highly promising for analyzing explanation data, by identifying the individual units (e.g., brain regions) of the learned mappings between brain activity and decoded cognitive states as well as the dynamics of their interactions.
Box 2. Recommended XAI approaches for cognitive decoding.

We generally recommend the application of backward decompositions (see Box 1 and [13-19,21,24]) to analyze the contribution of individual input features to the cognitive decoding decisions of DL models. Below, we provide three specific recommendations for XAI approaches for cognitive decoding:

**LRP** [13]: An overview of LRP is provided in Box 1. While several rules have been proposed to redistribute relevance $r$ between units $i$ and $j$ of two successive layers [13,49,50], recent empirical work has shown that a composite of these rules is best-suited for CV models [49,50]. Specifically, this work suggest combining the LRP-0 rule $(r_i = \sum_j \frac{a_{ij}}{\sum \sigma_j} r_j)$, where $a$ and $w$ represent the input and weights of unit $i$) for layers closer to the output, with the LRP-$\varepsilon$ rule $(r_i = \sum_j a_{ij}^\varepsilon r_j)$ with $1e^{-4} \leq \varepsilon < 1$) for middle layers, and the LRP-$\gamma$ rule $(r_i = \sum_j a_{ij}(\gamma w_j r_j))$, where $\gamma$ controls positive contributions and is generally $0 < \gamma$) for layers closer to the input. A TensorFlow implementation of LRP is provided by iNNvestigate [54], while Captum [55] provides PyTorch implementation.

**Deep learning important features (DeepLIFT)** [16]: DeepLIFT seeks to explain the contribution of model unit $i$ to prediction $f(x)$ by comparing the unit’s current activation $a_i$ to a reference activation $a_i^0$, which results from a reference input $x^0$[16]. The reference input $x^0$ should be neutral and preserve only those properties of the input that are not considered relevant to the decoding problem while removing properties that are considered relevant (e.g., for MNIST, an all-zero reference is recommended, as this is the MNIST background). For functional neuroimaging data, we thus recommend an all-zero reference, a reference that adds noise to the input as well as a reference involving samples from other decoding classes (e.g., their average). We further recommend using multiple references to increase robustness towards specific reference choices [23]. A TensorFlow implementation of DeepLIFT can be found at github.com/kundajelab/deeplift and github.com/slundberg/shap, while Captum [55] provides a PyTorch implementation.

**Integrated gradients (IG)** [15]: The integrated gradients (IG) technique is closely related to both LRP and DeepLIFT [56]. IG is applicable to any differentiable model and defines relevance $r_d$ of input feature $d$ for prediction $f(x)$ by integrating the gradient $f'(x)$ along some trajectory in the input space connecting a reference point $x^0$ to the current input $x$: $r_d = (x_d - x_d^0) \int_{a=0}^{1} \frac{\delta f(x^0 + a(x-x^0))}{\delta x_d} da$. Similar to DeepLIFT, the authors recommend an all-zero reference or the addition of noise to the input [14,57], while an average over multiple references is also possible [57]. A tutorial on how to use IG in TensorFlow can be found at tensorflow.org/tutorials/interpretability/integrated_gradients, while Captum [55] provides a PyTorch implementation.
Leveraging public data

A second major challenge for DL models in functional neuroimaging research is the high dimensionality and low sample size of conventional functional neuroimaging datasets. A typical functional Magnetic Resonance Imaging (fMRI) dataset contains a few hundred volumes for each of tens to hundreds of individuals, while each volume contains several hundred thousand voxels (or dimensions). Current state-of-the-art DL models, in contrast, can easily contain many hundred million parameters [58,59], while recent language models have pushed this boundary even further with many billion parameters [60-62]. In most cases, DL models thus contain many more trainable parameters than there are samples in their training data. While this vast overparameterization represents a key element to the empirical success of DL models, by enabling them to find near-perfect solutions for most standard learning tasks [63,64] and to generalize well between datasets [62,65,66], it also represents one of the biggest challenges for their application in fields where data are scarce, as the performance of DL models is strongly dependent on the amount of available training data [61,67].

To approach this challenge, researchers have developed many methods that aim to improve the performance of DL methods in smaller datasets (e.g., [68-71]). One of these methods, with strong empirical success, is transfer learning [71]. The goal of transfer learning is to leverage the knowledge about a mapping between input data and a target variable that can be learned from one dataset (i.e., the source or upstream domain) to subsequently improve the learning of a similar mapping in another dataset of a related domain (i.e., the target or downstream domain). Knowledge is typically transferred in the form of the parameters that a model has learned in the source domain and that are then used to initialize the model (or parts of the model) when beginning learning in the target domain. Transfer learning has been especially successful in CV and NLP, where large publicly available datasets exist (e.g., [72,73] and http://www.commoncrawl.org). Here, DL models are first pre-trained on these large datasets (e.g., to classify objects in images) and subsequently fine-tuned on smaller datasets of a related target domain (e.g., to classify brain tumors in medical imaging; [74]). Computationally, pre-training can aid subsequent optimizations by placing the parameters near a local minimum of the loss function [75] and by acting as a regularizer [76]. Pre-trained models generally exhibit faster learning and higher predictive accuracy, while also requiring less training data when compared to models that are trained from
scratch [61,65,66,71,75-78]. The benefits of pre-training can diminish, however, with increasing
size of the target dataset [61] and as the overall differences between source and target learning task
and/or domain increase [79,80].

Over recent years, the field of functional neuroimaging has experienced a similar increase
in the availability of public datasets, which are provided by large neuroimaging initiatives as well
as individual researchers [81]. In addition, several efforts have been made to standardize the
organization (e.g., [82,83]) and preprocessing (e.g., [84]) of functional neuroimaging data. These
developments have paved the way for the field of functional neuroimaging to enter a big data era,
allowing for the application of transfer learning.

First empirical evidence indicates that transfer learning between individuals [85-87],
experiment tasks [36, 37, 39,88], and datasets [89-92] is possible and that pre-training generally
improves the performance of DL models when applied to conventional fMRI datasets
[36,85,86,88,89,91]. Most of this work has utilized traditional supervised learning techniques
during pre-training, by assigning a cognitive state to each sample in the data and training a
decoding model to identify these states from the data (see Box 3). While this is a fruitful approach
to decoding analyses within individual neuroimaging datasets, it is often difficult to extend to
analyses across many datasets. In spite of several attempts (e.g., [93,94]), functional neuroimaging
research has yet to widely adopt standardized definitions of cognitive states. Without this type of
standardization, it is often unclear whether two experiments from two separate laboratories elicit
the same or different sets of cognitive states. Imagine the following experiments: In the first,
participants read aloud a sequence of sentences and are then asked to repeat the last word of each
sentence [95]. In the second, participants first hear a sequence of letters and digits and are then
asked to report back the letters and digits in alphabetical and numerical order respectively (the
letter–number sequencing task; [96]). While both experiments label the associated cognitive state
as “working memory”, one could argue that the experiments in fact elicit two distinct cognitive
states, as one solely requires temporarily storing information while the other also requires active
manipulation of this information.

This problem of imprecisely labeled data has been similarly encountered in machine
learning research. Here, researchers have developed weakly supervised learning techniques, which
aim to train DL models with noisy or incomplete data labels [97]. One prominent weak supervision
approach is data programming (see Box 3 and [98,99]), which seeks to alleviate the cost of
manually labeling large datasets by the use of simple labeling functions. These functions
automatically label subsets of the data by implementing simple domain heuristics of subject matter
experts (e.g., label a YouTube text comment as Spam if it contains a URL or the words “check this
out”). The generated labels are then used to train DL models in a supervised manner. Importantly,
data programming frameworks account for noise and conflicts that can arise from the automatic
labeling by representing the labeling process as a generative model [98,99]. Recent empirical work
has demonstrated that this type of weak supervision can be successfully used for the classification
of unlabeled medical imaging data (e.g., radiography or computer tomography data; [100]), by
designing labeling functions that extract labels from the accompanying medical text reports. A
similar approach could be fruitful to generate standardized labels of cognitive states (e.g.,
according to the Cognitive Atlas; [94]) by applying these types of automatic labeling functions to
the accompanying publication texts (e.g., label an fMRI scan as “visual perception” if the
publication text contains the words “viewed” or “viewing” in the Methods section).

An alternative approach, with strong recent empirical success, is unsupervised learning
(see Box 3 and [101,102]). Unsupervised learning does not consider any labeling and instead trains
models to autonomously learn meaningful representations of the data. These learned
representations can then be used to improve learning in a related target domain. Two prominent
examples of unsupervised learning are contrastive and generative learning [103]. Both learn a
representation of the data by training an encoder model to project the data into a lower-dimensional
representation, which preserves relevant information. In contrastive learning [104], the encoder
model is trained by the use of an additional discriminator model, which aims to determine the
similarity of a pair of data samples based on their projection through the encoder model. During
training, positive pairs of samples are created by randomly augmenting the same data sample twice
(e.g., different views of the same image), while negative pairs are created by augmenting two
different data samples. Generative learning [105], in contrast, trains the encoder model by the use
of an additional decoder model, which aims to reconstruct the original data sample from the lower-
dimensional representation of the encoder model.
Box 3. Approaches to pre-training.

Transfer learning aims to improve the performance of model \( f(\cdot) \) in a target learning task \( T_T \) in a target domain \( D_T \) by leveraging knowledge that can be learned by \( \text{pre-training} f(\cdot) \) in a related source learning task \( T_S \) and source domain \( D_S \) [71]. Knowledge is generally transferred through a set of model weights \( W \) that \( f(\cdot) \) has learned during pre-training. A domain \( D \) is defined by a feature space \( X \) with samples \( x \in \mathbb{R}^N \) whose \( N \) feature values are characterised by some probability distribution \( P(X) \). Given a domain, a learning task \( T \) consists of learning some mapping between a label space \( Y \) and the associated feature space \( X \), by training \( f(\cdot) \) to accurately predict the target values \( y \in Y \) assigned to the samples \( x \in X \) in a dataset \( A = \{(x_0, y_0), \ldots, (x_m, y_m)\} \), such that \( f(x_i) \approx y_i \). Here, we describe three learning tasks (Fig. I), which enable \( f(\cdot) \) to learn in a source domain, when the target values \( y \) of a source dataset \( A_S \) are either fully accessible, noisy or incomplete, or not accessible.

**Supervised learning** [106]: In supervised learning, the target values \( y \in A_S \) are fully accessible and can be used to train \( f(\cdot) \), for example, by minimizing the distance between \( f(x) \) and \( y^*: \min_W \sum_{(x_i, y_i) \in A_S} (y_i - f(x_i))^2 \).

**Weakly supervised learning** [97]: In weakly supervised learning, the target values \( y \in A_S \) are noisy or incomplete. A prominent example of weak supervision is data programming [98], where noisy target values \( \hat{y} \) are generated for an unlabeled source dataset by the use of user-specified labeling functions. These labeling functions implement domain heuristics of subject matter experts (e.g., label a chest radiograph as “abnormal” if the corresponding medical text report contains a word with the prefix “pneumo”; [100]). The generated target values are then used to train \( f(\cdot) \) in a supervised way.

**Unsupervised learning** [101,102]: In unsupervised learning, the target values \( y \in A_S \) are not accessible. Instead, a new learning task is devised, which requires \( f(\cdot) \) to independently learn a representation of the data in the source domain. Two prominent unsupervised learning strategies are contrastive and generative learning. Both treat \( f(\cdot) \) as an encoder model, which is trained to project the samples \( x \in A_S \) into a lower-dimensional representation that preserves relevant information: \( f(\cdot): \mathbb{R}^N \rightarrow \mathbb{R}^L \) (where \( L < N \)). In contrastive learning [104], \( f(\cdot) \) is trained by the use of an additional discriminator model \( d(\cdot): \mathbb{R}^L \rightarrow \mathbb{R} \), which learns to determine the similarity of a pair of data samples based on the encoder’s projection. During training, a set of augmented versions \( \{x\} \) of the data samples \( x \in A_S \) is created and the discriminator’s task is to identify pairs \( \{\hat{x}_k, \hat{x}_j\} \) that result from the same sample \( x_i \). In generative learning [105], \( f(\cdot) \) is trained by the use of an additional decoder model \( d(\cdot): \mathbb{R}^L \rightarrow \mathbb{R}^N \), which aims to reconstruct the original data sample from the encoder’s projection: \( d(f(x_i)) \approx x_i \).
Ensuring reproducibility and robustness

Recent work in functional neuroimaging has exposed the high flexibility of its standard analysis workflows, leading to substantial variability in results and scientific conclusions [107]. In light of these issues, various efforts have been made to improve the standardization and reproducibility of functional neuroimaging analyses (e.g., [82,84]). DL research is currently facing similar concerns, with model performances that are often hard to reproduce [108-111] and not robust to the diversity of real-word data [79,112-122]. Functional neuroimaging researchers who are interested in applying DL methods to cognitive decoding analyses are thus faced with additional challenges for the reproducibility and robustness of their work, which arise at the intersection of both fields.

Methodological progress in DL research is often driven by a hunt for state-of-the-art performances in benchmarks (see http://www.paperswithcode.com/sota), that is, by whether a new methodology outperforms existing ones in pre-defined test datasets. While this approach has helped the field of DL to evolve fast and quickly develop accurate models, it has also established a research culture that often sacrifices scientific rigor for maximal performance metrics [123,124], not unlike the “p-hacking” phenomenon in null hypothesis testing [125].

A central argument for predefined test datasets is that all models should be compared on the same grounds (i.e., the same sets of training and testing samples). Yet, these types of point estimates are often insufficient to determine whether a model actually outperforms others in new data. Recent empirical work has demonstrated, for example, that the convergence of DL models and thereby their final performance in a test dataset is dependent on many non-deterministic factors
of the training, such as random weight initializations and random shufflings or augmentations of
the data during training [110,111,126], as well as the specific choices for hyper-parameters, such
as the specification of individual model layers and optimization algorithm [111,127,128]. In some
cases, researchers can thus achieve state-of-the-art performance simply by investing large
computational budgets into tuning these types of factors for a specific test dataset. Consequently,
many of the currently reported DL benchmarks are built on top of massive computational budgets
and are often difficult to reproduce by other researchers without full access to the original code
and computing environments [110,111,126,129]. Recent empirical findings further suggest that
the comparisons performed on several of these benchmarks lack the statistical power required to
accurately determine the reported improvements in model performance [130], a problem similarly
evident in neuroimaging research [131].

In addition to these challenges for the reproducibility of benchmark model performances,
recent findings have also demonstrated that the resulting highly tuned models often lack basic
robustness towards slight distributional shifts [112,113,115-117,132] or corruptions [121,133] of
the data, while model performances can also vary widely across the different subpopulations of a
dataset (stratified, for example, by people’s age, race or gender; [134]) or be based in spurious
shortcuts that the models have learned from a training dataset but which do not generalize well to
other scenarios [52,118].

For these reasons, researchers have started advocating for more comprehensive and
standardized reporting of the training history of DL models [135], more extensive evaluation
procedures [112-115,136], as well as an increased scientific rigor in DL research [123]. In the
following, we briefly outline several recommendations resulting from this work, which aim to
improve the reproducibility and robustness of model performances (for a brief overview, see Box
4).

Most DL training pipelines are too complex to allow for a comprehensive evaluation of all
possible choices of hyper-parameters and other non-deterministic factors of the training. However,
evaluating only a specific instance of these choices does not give a reliable estimate of a model’s
expected performance in new data. Instead, recent empirical work suggests to randomize as many
of these choices as possible over multiple training runs, given the computational budget at hand
[136]. This allows to better account for the variance in model performance associated with these
choices and thus to obtain a more accurate estimate of the model’s expected performance (see [126,135,136]).

In addition, researchers have advocated for the use of multiple random splits of the data into training, validation, and test datasets to evaluate model performances (e.g., with bootstrapping or cross-validation; [109,136]). A single, predefined test dataset generally contains limited information about the whole underlying data distribution and is thus also limited in its ability to provide an accurate estimate of the model’s expected performance in new data. Multiple random splits, in contrast, better account for the variance in model performance associated with different splits of the data, allowing to reduce the error in a model’s expected performance.

Further, to ensure that the chosen combination of statistical comparison method and test dataset size provides sufficient statistical power to accurately determine the studied difference in model performance, researchers have recently suggested the use of simple simulation studies, by first identifying and estimating the required quantities of the statistical testing procedure (e.g., McNemar’s test for paired data requires the models’ probabilities of making a correct prediction as well as their agreement rate), and subsequently using these estimates to simulate model comparisons at different dataset sizes [130]. In addition to ensuring that the chosen performance evaluation procedure does not lack statistical power, recent work in neuroimaging also suggests to control for multiple sequential model comparisons, as multiple sequential hypothesis tests (e.g., performance comparisons) on the same dataset can inflate false positive rates [137].

Next to choosing appropriate means of evaluating model performances in a given dataset, it is essential to ensure that the resulting performances are robust towards real-world data (e.g., neuroimaging data from different individuals, tasks, and laboratories). Model performances can falter, for example, when encountering slight distributional shifts or corruptions of the data, as recently demonstrated by researchers who confronted benchmark DL models with datasets that mimicked the original benchmarks but included new samples from the original data sources (thus introducing distributional shifts; [79]) or by corrupting the inputs with simple noise [121,133]. To strengthen robustness towards these types of scenarios, empirical work suggests the use of random data augmentations during training, for example, by randomly flipping images from left-to-right [138] or by randomly occluding [139] or mixing [140,141] inputs. It can also be beneficial to mix different data augmentation strategies [142] or to automatically learn these strategies from the data [143].
Similarly, DL model performances have been shown to often vary highly across the different, often unrecognized, subpopulations of a dataset (a phenomenon known as “hidden stratification”; [119,120]). For example, a DL model trained to distinguish benign and malignant tumors from medical imaging data can perform well on average, while consistently misclassifying an aggressive but rare form of cancer. To detect and measure these types of hidden stratification, recent work suggests three approaches [119]: In schema completion, domain experts define and provide a comprehensive set of subclasses for the test dataset (e.g., a more comprehensive labeling of the cognitive states at different time points of the experiment), which can then be used to evaluate the performance of models trained on the more coarsely-labeled training data. In error auditing, domain experts retrospectively inspect instances of false model predictions to identify possible hidden subpopulations (e.g., in combination with XAI techniques; see Box 1). In automatic algorithmic approaches, dedicated algorithms independently search for hidden subpopulations in the data and/or model predictions (e.g., by the use of clustering techniques). Once hidden stratification is detected, modern DL techniques can be used to improve model performances on subpopulations (e.g., [144]).

Lastly, it was demonstrated that DL models can be susceptible to learning spurious shortcuts that allow them to perform well in a given training dataset but which do not generalize well to other scenarios [52,118,122]. Researchers found, for example, that a pneumonia detection model trained with medical imaging data can learn to perform well on average solely by learning to identify hospital-specific artifacts in the medical images in addition to learning the hospitals’ pneumonia prevalence rates [145]. To identify these types of shortcuts, researchers recommend to evaluate model performances on out-of-distribution data, for example, on data from different sources (e.g., neuroimaging data from different laboratories and individuals), and to inspect instances of the data whenever out-of-distribution error rates are high relative to in-distribution errors (e.g., with visual inspection and/or the application of XAI techniques; see Box 1).
Box 4. Recommendations to improve the reproducibility and robustness of DL modelling results in cognitive decoding.

The performances of DL models in benchmarks are often difficult to reproduce by other researchers or in new data, as the convergence of DL models (and thereby their final performance) is strongly dependent on many non-deterministic aspects of the training [110,111,135,136]. Further, the resulting highly tuned benchmark performances are often not robust towards the diversity of real-world data [79,112-116]. Below, we provide a set of general recommendations to improve the reproducibility and robustness of DL model performances in cognitive decoding analyses:

- Use multiple training runs to estimate a model’s expected performance, given the computational budget at hand (for methodological suggestions on how to estimate a model’s expected performance (or to perform model comparisons) based on multiple training runs, see [126,135,136]).
- Apply random augmentations to the data during training, such as random occlusions [139] and combinations [140,141] of the inputs.
- For each training run, randomize as many aspects of the training pipeline as possible (including random seeds, random weight initializations, and random shufflings of the training data) and use a random split of the data into training, validation, and test datasets (e.g., by the use of bootstrapping or cross-validation; [109,136]).
- If model comparisons are performed, ensure that the chosen combination of statistical comparison procedure and test dataset size has enough statistical power to accurately determine the studied differences in model performance (e.g., by the use of simple simulation studies; see [130]).
- Whenever possible, evaluate model performances on out-of-distribution data (e.g., by using neuroimaging data from different laboratories and individuals) and test for hidden stratification [119,120] by the use of schema completion, error auditing or automatic algorithmic approaches (for methodological details, see [119]).
- Finally, publicly share any used data, analysis code, and computing environment (e.g., by the use of containerization with Docker or Singularity) in a dedicated repository (e.g., Open Science Framework; [146]).

Concluding remarks

DL methods have experienced great success in research and industry and have had major impacts on society [1]. This success has triggered interest in their application to the field of cognitive decoding, where researchers aim to characterize the representations of different brain regions by identifying the set of cognitive states that can be accurately decoded (or identified) from the activity of these regions. DL methods hold a high promise to revolutionize cognitive decoding analyses with their unmatched ability to learn versatile representations of complex data. Yet, fully
leverage the potential of DL methods in cognitive decoding is currently hindered by three main challenges, which result from a general lack of interpretability of DL methods as well as difficulties in applying them to small datasets and in ensuring their reproducibility and robustness.

Here, we have provided a detailed discussion of these three challenges and proposed a set of solutions that are informed by recent advances in functional neuroimaging and machine learning research. In sum, we recommend that researchers utilize XAI techniques to identify the mapping between cognitive states and brain activity that a DL model has learned (Box 1-2), improve the performance of DL methods in conventional neuroimaging datasets by pre-training these models on public neuroimaging data (Box 3), and follow a set of specific recommendations to improve the reproducibility and robustness of DL model performances (Box 4). We hope that researchers will take inspiration from our discussion and explore the many open research questions that remain on the path to determining whether DL methods can live up to their promise for cognitive decoding.

Acknowledgments

Armin W. Thomas is supported by Stanford Data Science through the Ram and Vijay Shriram Data Science Fellowship. Russell A. Poldrack is supported by the National Science Foundation under Grant No. OAC-1760950. Christopher Ré gratefully acknowledges the support of NIH under No. U54EB020405 (Mobilize), NSF under Nos. CCF1763315 (Beyond Sparsity), CCF1563078 (Volume to Velocity), and 1937301 (RTML); ONR under No. N000141712266 (Unifying Weak Supervision); the Moore Foundation, NXP, Xilinx, LETI-CEA, Intel, IBM, Microsoft, NEC, Toshiba, TSMC, ARM, Hitachi, BASF, Accenture, Ericsson, Qualcomm, Analog Devices, the Okawa Foundation, American Family Insurance, Google Cloud, Salesforce, Total, the HAI-AWS Cloud Credits for Research program, Stanford Data Science, and members of the Stanford DAWN project: Facebook, Google, and VMware. The Mobilize Center is a Biomedical Technology Resource Center, funded by the NIH National Institute of Biomedical Imaging and Bioengineering through Grant P41EB027060. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this material are those
of the authors and do not necessarily reflect the views, policies, or endorsements, either expressed or implied, of NIH, ONR, or the U.S. Government.

References

[1] Goodfellow, I. et al. (2016). Deep learning. MIT press.

[2] Livezey, J. A. and Glaser, J. I. (2021). Deep learning approaches for neural decoding across architectures and recording modalities. Briefings in bioinformatics, 22(2), 1577-1591. doi: 10.1093/bib/bbaa355.

[3] Norman, K. A. et al. (2006). Beyond mind-reading: multi-voxel pattern analysis of fMRI data. Trends in cognitive sciences, 10(9), 424-430. doi: 10.1016/j.tics.2006.07.005.

[4] Poldrack, R. A et al. (2015). Long-term neural and physiological phenotyping of a single human. Nature communications, 6(1), 1-15. doi: 10.1038/ncomms9885.

[5] Tavor, I., et al. (2016). Task-free MRI predicts individual differences in brain activity during task performance. Science, 352(6282), 216-220. doi: 10.1126/science.aad8127.

[6] Cole, M. W. et al. (2014). Intrinsic and task-evoked network architectures of the human brain. Neuron, 83(1), 238-251. doi: 10.1016/j.neuron.2014.05.014.

[7] Gunning, D. et al. (2019). XAI—Explainable artificial intelligence. Science Robotics, 4(37). doi: 10.1126/scirobots.aay7120.

[8] Samek, W et al. (2021). Explaining deep neural networks and beyond: A review of methods and applications. Proceedings of the IEEE, 109(3), 247-278. doi: 10.1109/JPROC.2021.3060483.

[9] Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608. Available: http://arxiv.org/abs/1702.08608

[10] Montavon, G. et al. (2018). Methods for interpreting and understanding deep neural networks. Digital Signal Processing, 73, 1-15. doi: 10.1016/j.dsp.2017.10.011.

[11] Nguyen, A. et al. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. Advances In Neural Information Processing Systems, 29, 3387-3395. Available: http://papers.nips.cc/paper/6519-synthesizing-the-preferred-inputs-for-neurons-in-neural-networks-via-deep-generator-networks.pdf

[12] Simonyan, K. et al. (2013). Deep inside convolutional networks: Visualising image
classification models and saliency maps. arXiv preprint arXiv:1312.6034. Available: http://arxiv.org/abs/1312.6034

[13] Bach, S. et al. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one, 10(7), e0130140. doi: 10.1371/journal.pone.0130140.

[14] Smilkov, D. et al. (2017). Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825. Available: http://arxiv.org/abs/1706.03825

[15] Sundararajan, M. et al. (2017). Axiomatic attribution for deep networks. In International Conference on Machine Learning (ICML), 70, 3319-3328. PMLR. Available: http://proceedings.mlr.press/v70/sundararajan17a.html

[16] Shrikumar, A. et al. (2017). Learning important features through propagating activation differences. In International Conference on Machine Learning (ICML), 70, 3145-3153. PMLR. Available: http://proceedings.mlr.press/v70/shrikumar17a.html

[17] Springenberg, J. T. et al. (2014). Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806. Available: http://arxiv.org/abs/1412.6806

[18] Selvaraju, R. R. et al. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. IEEE international conference on computer vision (ICCV), 618-626. Available: https://openaccess.thecvf.com/content_iccv_2017/html/Selvaraju_Grad-CAM_Visual_Explanations_ICCV_2017_paper.html

[19] Kindermans, P. J. et al. (2017). Learning how to explain neural networks: Patternnet and patternattribution. arXiv preprint arXiv:1705.05598. Available: http://arxiv.org/abs/1705.05598

[20] Fong, R. C. and Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. IEEE international conference on computer vision (ICCV), 3429-3437. Available: https://openaccess.thecvf.com/content_iccv_2017/html/Fong_Interpretable_Explanations_of_ICCV_2017_paper.html

[21] Montavon, G. et al. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition, 65, 211-222. doi: 10.1016/j.patcog.2016.11.008.

[22] Zurada, J. M. et al. (1994). Sensitivity analysis for minimization of input data dimension for
feedforward neural network. IEEE International Symposium on Circuits and Systems-ISCAS'94, 6, 447-450 doi: 10.1109/ISCAS.1994.409622.

[23] Lundberg, S. M. and Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 4768-4777. Available: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43d6d28b67767-Paper.pdf

[24] Zeiler M.D. and Fergus R. (2014) Visualizing and Understanding Convolutional Networks. In: Fleet D. et al. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8689. Springer, Cham. https://doi.org/10.1007/978-3-319-10590-1_53

[25] Zintgraf, L. M. et al. (2017). Visualizing deep neural network decisions: Prediction difference analysis. arXiv preprint arXiv:1702.04595. Available: http://arxiv.org/abs/1702.04595

[26] Ribeiro, M. T. et al. (2016). "Why should i trust you?" Explaining the predictions of any classifier. ACM SIGKDD international conference on knowledge discovery and data mining, 1135-1144. doi: 10.1145/2939672.2939778.

[27] Luong, T. et al. (2015). Effective Approaches to Attention-based Neural Machine Translation. Conference on Empirical Methods in Natural Language Processing (EMNLP), 1412–1421. doi: 10.18653/v1/D15-1166.

[28] Xu, K. et al. (2015). Show, attend and tell: Neural image caption generation with visual attention. International conference on machine learning (ICML), 37, 2048-2057. PMLR. Available: http://proceedings.mlr.press/v37/xuc15.html

[29] Vaswani, A. et al. (2017). Attention is all you need. Advances In Neural Information Processing Systems, 30, 5998-6008. Available: https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[30] Kim, B. et al. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). International conference on machine learning (ICML), 80, 2668-2677. PMLR. Available: http://proceedings.mlr.press/v80/kim18d.html

[31] Alvarez-Melis, D. and Jaakkola, T. S. (2018, December). Towards robust interpretability with self-explaining neural networks. Advances in Neural Information Processing Systems,
Dinsdale, N. K. et al. (2021). Learning patterns of the ageing brain in MRI using deep convolutional networks. Neuroimage, 224, 117401. doi: 10.1016/j.neuroimage.2020.117401.

Oh, K. et al. (2019). Classification and visualization of Alzheimer’s disease using volumetric convolutional neural network and transfer learning. Scientific Reports, 9(1), 1-16. doi: 10.1038/s41598-019-54548-6.

Thomas, A. W. et al. (2019). Analyzing neuroimaging data through recurrent deep learning models. Frontiers in neuroscience, 13, 1321. doi: 10.3389/fnins.2019.01321.

Kohoutová, L. et al. (2020). Toward a unified framework for interpreting machine-learning models in neuroimaging. Nature protocols, 15(4), 1399-1435. doi: 10.1038/s41596-019-0289-5.

Zhang, Y. et al. (2021). Functional annotation of human cognitive states using deep graph convolution. NeuroImage, 231, 117847. doi: 10.1016/j.neuroimage.2021.117847.

Wang, X. et al. (2020). Decoding and mapping task states of the human brain via deep learning. Human brain mapping, 41(6), 1505-1519. doi: https://doi.org/10.1002/hbm.24891.

Hu, J. et al. (2021). Deep learning-based classification and voxel-based visualization of frontotemporal dementia and Alzheimer’s disease. Frontiers in Neuroscience, 14, 1468. doi: 10.3389/fnins.2020.626154.

Nguyen, S. et al. (2020). Attend and Decode: 4D fMRI Task State Decoding Using Attention Models. Machine Learning for Health, 136, 267-279. PMLR. Available: http://proceedings.mlr.press/v136/nguyen20a.html

Zhang, T. et al. (2020). Separated channel attention convolutional neural network (SC-CNN-attention) to identify ADHD in multi-site rs-fMRI dataset. Entropy, 22(8), 893. doi: 10.3390/e22080893.

Petersen, S. E. and Posner, M. I. (2012). The attention system of the human brain: 20 years after. Annual review of neuroscience, 35, 73-89. doi: 10.1146/annurev-neuro-062111-150525.

Jain, S. and Wallace, B. C. (2019). Attention is not explanation. arXiv preprint
[43] Serrano, S. and Smith, N. A. (2019). Is Attention Interpretable?. Annual Meeting of the Association for Computational Linguistics, 2931-2951. doi: 10.18653/v1/P19-1282.

[44] Samek, W. et al. (2016). Evaluating the visualization of what a deep neural network has learned. IEEE transactions on neural networks and learning systems, 28(11), 2660-2673. doi: 10.1109/TNNLS.2016.2599820.

[45] Kindermans, P. J. et al. (2019). The (un)reliability of saliency methods. Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, 267-280. Springer, Cham. doi: 10.1007/978-3-030-28954-6_14.

[46] Sun, J. et al. (2020). Understanding image captioning models beyond visualizing attention. arXiv preprint arXiv:2001.01037. Available: http://arxiv.org/abs/2001.01037

[47] Adebayo, J. et al. (2018). Sanity Checks for Saliency Maps. Advances in Neural Information Processing Systems, 31, 9505-9515. Available: https://proceedings.neurips.cc/paper/2018/file/294a8ed24b1ad22ec2e7eafa049b8737-Paper.pdf

[48] Jacovi, A. and Goldberg, Y. (2020). Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness?. arXiv preprint arXiv:2004.03685. Available: http://arxiv.org/abs/2004.03685

[49] Kohlbrenner, M. et al. (2020). Towards best practice in explaining neural network decisions with LRP. International Joint Conference on Neural Networks (IJCNN), 1-7. IEEE. doi: 10.1109/IJCNN48605.2020.9206975.

[50] Montavon G. et al. (2019) Layer-Wise Relevance Propagation: An Overview. Samek, W. et al. (eds) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science, 11700. Springer, Cham. doi: 10.1007/978-3-030-28954-6_10

[51] Hanson, S. J. et al. (2004). Combinatorial codes in ventral temporal lobe for object recognition: Haxby (2001) revisited: is there a “face” area?. Neuroimage, 23(1), 156-166. doi: 10.1016/j.neuroimage.2004.05.020.

[52] Lapuschkin, S. et al. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. Nature communications, 10(1), 1-8. doi: 10.1038/s41467-019-08987-4.

[53] Bassett, D. S. and Sporns, O. (2017). Network neuroscience. Nature neuroscience, 20(3),
[54] Alber, M. et al. (2019). iNNvestigate neural networks!. J. Mach. Learn. Res., 20(93), 1-8.
[55] Kokhlikyan, N. et al. (2020). Captum: A unified and generic model interpretability library for pytorch. arXiv preprint arXiv:2009.07896. Available: http://arxiv.org/abs/2009.07896
[56] Ancona, M. et al. (2018). Towards better understanding of gradient-based attribution methods for Deep Neural Networks. International Conference on Learning Representations. doi: 10.5167/uzh-168590.
[57] Sturmfels, P. et al. (2020). Visualizing the impact of feature attribution baselines. Distill, 5(1), e22. doi: 10.23915/distill.00022.
[58] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. Available: http://arxiv.org/abs/1409.1556
[59] Devlin, J. et al. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. Available: http://arxiv.org/abs/1810.04805
[60] Zeng, W. et al. (2021). PanGu-alpha$: Large-scale Autoregressive Pretrained Chinese Language Models with Auto-parallel Computation. arXiv preprint arXiv:2104.12369. Available: http://arxiv.org/abs/2104.12369
[61] Raffel, C. et al. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research, 21, 1-67.
[62] Brown, T. et al. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33, 1877-1901. Available: https://papers.nips.cc/paper/2020/file/1457c0d6bfcf4967418b8ac142f6a-Paper.pdf
[63] Allen-Zhu, Z. et al. (2019) A convergence theory for deep learning via over-parameterization. International Conference on Machine Learning (ICML), 97, 242-252. PMLR. Available: http://proceedings.mlr.press/v97/allen-zhu19a.html
[64] Li, Y. and Liang, Y. (2018). Learning overparameterized neural networks via stochastic gradient descent on structured data. Advances in Neural Information Processing Systems, 8168-8177. https://proceedings.neurips.cc/paper/2018/file/54fe976ba170c19ebae43679b362263-Paper.pdf
[65] Kolesnikov A. et al. (2020) Big Transfer (BiT): General Visual Representation Learning.
Vedaldi, A. et al. (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in
Computer Science, 12350. Springer, Cham. https://doi.org/10.1007/978-3-030-58558-7_29
[66] Chen, T. et al. (2020). Big Self-Supervised Models are Strong Semi-Supervised Learners.
Advances in Neural Information Processing Systems, 33, 22243-22255. Available:
https://proceedings.neurips.cc/paper/2020/file/fcbe95ccdd551da181207c0c1400c655-
Paper.pdf
[67] Sun, C. et al. (2017). Revisiting unreasonable effectiveness of data in deep learning era.
IEEE international conference on computer vision (ICCV), 843-852. doi:
10.1109/ICCV.2017.97.
[68] Krogh, A. and Hertz, J. A. (1992). A simple weight decay can improve generalization.
Advances In Neural Information Processing Systems, 950-957. Available:
http://papers.nips.cc/paper/563-a-simple-weight-decay-can-improve-generalization.pdf
[69] Pascanu, R. et al. (2013). On the difficulty of training recurrent neural networks.
International conference on machine learning (ICML), 28, 1310-1318. PMLR. Available:
http://proceedings.mlr.press/v28/pascanu13.html
[70] Srivastava, N. et al. (2014). Dropout: a simple way to prevent neural networks from
overfitting. The journal of machine learning research, 15(1), 1929-1958.
[71] Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on
knowledge and data engineering, 22(10), 1345-1359. doi: 10.1109/TKDE.2009.191.
[72] Deng, J. et al. (2009). ImageNet: A large-scale hierarchical image database. IEEE
Conference on Computer Vision and Pattern Recognition (CVPR), 248-255. IEEE. doi:
10.1109/CVPR.2009.5206848.
[73] Bowman, S. R. et al. (2015). A large annotated corpus for learning natural language
inference. Conference on Empirical Methods in Natural Language Processing (EMNLP),
632-642. doi: 10.18653/v1/D15-1075.
[74] Deepak, S. and Ameer, P. M. (2019). Brain tumor classification using deep CNN features
via transfer learning. Computers in biology and medicine, 111, 103345. doi:
10.1016/j.compbiomed.2019.103345.
[75] Bengio, Y. et al. (2007). Greedy layer-wise training of deep networks. Advances in neural
information processing systems, 19, 153-160. Available:
[76] Erhan, D. et al. (2010). Why does unsupervised pre-training help deep learning?. International conference on artificial intelligence and statistics, 9, 201-208. Available: http://proceedings.mlr.press/v9/erhan10a.html

[77] Yosinski, J. et al. (2014). How transferable are features in deep neural networks?. Advances In Neural Information Processing Systems, 27, 3320-3328. Available: http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf

[78] Kornblith, S. et al. (2019). Do better imagenet models transfer better?. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2661-2671. doi: 10.1109/CVPR.2019.00277.

[79] Recht, B. et al. (2019). Do imagenet classifiers generalize to imagenet?. International Conference on Machine Learning (ICML), 97, 5389-5400. PMLR. Available: http://proceedings.mlr.press/v97/recht19a.html

[80] He, K. et al. (2019). Rethinking imagenet pre-training. IEEE/CVF International Conference on Computer Vision (ICCV), 4918-4927. doi: 10.1109/ICCV.2019.00502.

[81] Horien, C. et al. (2021). A hitchhiker’s guide to working with large, open-source neuroimaging datasets. Nature human behaviour, 5(2), 185-193. doi: 10.1038/s41562-020-01005-4.

[82] Gorgolewski, K. J. et al. (2016). The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. Scientific data, 3(1), 1-9. doi: 10.1038/sdata.2016.44.

[83] Markiewicz, C. J. et al. (2021). OpenNeuro: An open resource for sharing of neuroimaging data. bioRxiv. doi: 10.1101/2021.06.28.450168.

[84] Esteban, O. et al. (2019). fMRIPrep: a robust preprocessing pipeline for functional MRI. Nature methods, 16(1), 111-116. doi: 10.1038/s41592-018-0235-4.

[85] Mahmood, U. et al. (2019). Transfer learning of fMRI dynamics. arXiv preprint arXiv:1911.06813. Available: http://arxiv.org/abs/1911.06813

[86] Koyamada, S. et al. (2015). Deep learning of fMRI big data: a novel approach to subject-transfer decoding. arXiv preprint arXiv:1502.00093. Available: http://arxiv.org/abs/1502.00093

[87] Zheng, W. L. et al. (2016). Personalizing EEG-based affective models with transfer
learning. International joint conference on artificial intelligence, 2732-2738.

[88] Thomas A.W. et al. (2019) Deep Transfer Learning for Whole-Brain FMRI Analyses.

Zhou L. et al. (eds) OR 2.0 Context-Aware Operating Theaters and Machine Learning in Clinical Neuroimaging. OR 2.0 2019, MLCN 2019. Lecture Notes in Computer Science, vol 11796. Springer, Cham. https://doi.org/10.1007/978-3-030-32695-1_7

[89] Zhang, H. et al. (2018). Transfer learning on fMRI datasets. International Conference on Artificial Intelligence and Statistics, 84, 595-603. PMLR. Available: http://proceedings.mlr.press/v84/zhang18b.html

[90] Yousefnezhad, M. et al. (2020). Shared Space Transfer Learning for analyzing multi-site fMRI data. Advances in Neural Information Processing Systems, 33, 15990-16000. https://proceedings.neurips.cc/paper/2020/file/b837305e43f7e535a1506fc263eee3ed-Paper.pdf

[91] Mensch, A. et al. (2021). Extracting representations of cognition across neuroimaging studies improves brain decoding. PLoS computational biology, 17(5), e1008795. doi: 10.1371/journal.pcbi.1008795.

[92] Zhou S. et al. (2019) Improving Whole-Brain Neural Decoding of fMRI with Domain Adaptation. Suk, H. I. et al. (eds) Machine Learning in Medical Imaging. MLMI 2019. Lecture Notes in Computer Science, vol 11861. Springer, Cham. https://doi.org/10.1007/978-3-030-32692-0_31

[93] Turner, J. A. and Laird, A. R. (2012). The cognitive paradigm ontology: design and application. Neuroinformatics, 10(1), 57-66. doi: 10.1007/s12021-011-9126-x.

[94] Poldrack, R. A. et al. (2011). The cognitive atlas: toward a knowledge foundation for cognitive neuroscience. Frontiers in neuroinformatics, 5, 17. doi: 10.3389/fninf.2011.00017.

[95] Daneman, M. and Carpenter, P. A. (1980). Individual differences in working memory and reading. Journal of verbal learning and verbal behavior, 19(4), 450-466. doi: 10.1016/S0022-4371(80)90312-6.

[96] Wechsler, D. (2008). Wechsler Adult Intelligence Scale--Fourth Edition. doi: 10.1037/t15169-000.

[97] Zhou, Z. H. (2018). A brief introduction to weakly supervised learning. National science review, 5(1), 44-53. doi: 10.1093/nsr/nwx106.
[98] Ratner, A. J. et al. (2016). Data programming: Creating large training sets, quickly. Advances In Neural Information Processing Systems, 29, 3567-3575. Available: https://proceedings.neurips.cc/paper/2016/file/6709e8d64a5f47269ed5cea9f625f7ab-Paper.pdf

[99] Ratner, A. et al. (2017). Snorkel: Rapid training data creation with weak supervision. VLDB Endowment. International Conference on Very Large Data Bases, 11(3), 269. NIH Public Access. doi: 10.14778/3157794.3157797.

[100] Dunnmon, J. A. et al. (2020). Cross-modal data programming enables rapid medical machine learning. Patterns, 1(2), 100019. doi: 10.1016/j.patter.2020.100019.

[101] Bengio, Y. et al. (2013). Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1798-1828. doi: 10.1109/TPAMI.2013.50.

[102] Barlow, H. B. (1989). Unsupervised learning. Neural computation, 1(3), 295-311. doi: 10.1162/neco.1989.1.3.295.

[103] Liu, X. et al. (2021). Self-supervised learning: Generative or contrastive. IEEE Transactions on Knowledge and Data Engineering. doi: 10.1109/TKDE.2021.3090866.

[104] Chen, T. et al. (2020). A simple framework for contrastive learning of visual representations. International conference on machine learning (ICML), 119,1597-1607. PMLR. Available: http://proceedings.mlr.press/v119/chen20j.html

[105] Hinton, G. E. et al. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554. doi: 10.1162/neco.2006.18.7.1527.

[106] Kotsiantis, S. B. et al. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), 3-24.

[107] Botvinik-Nezer, R. et al. (2020). Variability in the analysis of a single neuroimaging dataset by many teams. Nature, 582(7810), 84-88. doi: 10.1038/s41586-020-2314-9.

[108] Bouthillier, X., et al. (2019). Unreproducible research is reproducible. International Conference on Machine Learning (ICML), 97, 725-734. PMLR. Available: http://proceedings.mlr.press/v97/bouthillier19a.html

[109] Gorman, K. and Bedrick, S. (2019). We need to talk about standard splits. Association for computational linguistics, 2786-2791. doi: 10.18653/v1/P19-1267.
[110] Henderson, P. et al. (2018). Deep reinforcement learning that matters. AAAI conference on artificial intelligence, 32(1). Available: https://ojs.aaai.org/index.php/AAAI/article/view/11694

[111] Lucic, M. et al. (2018). Are GANs Created Equal? A Large-Scale Study. Advances In Neural Information Processing Systems, 31. Available: https://papers.nips.cc/paper/2018/file/e46de7e1bcaaced9a54f1e9d0d2f800d-Paper.pdf

[112] Koh, P. W. et al. (2021). Wilds: A benchmark of in-the-wild distribution shifts. International Conference on Machine Learning (ICML), 139, 5637-5664. PMLR. Available: http://proceedings.mlr.press/v139/koh21a.html

[113] Hendrycks, D. et al. (2020). The many faces of robustness: A critical analysis of out-of-distribution generalization. arXiv preprint arXiv:2006.16241. Available: http://arxiv.org/abs/2006.16241

[114] Kiela, D. et al. (2021). Dynabench: Rethinking Benchmarking in NLP. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 4110-4124. doi: 10.18653/v1/2021.naacl-main.324.

[115] Goel, K. et al. (2021). Robustness Gym: Unifying the NLP Evaluation Landscape. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations, 42-55. doi: 10.18653/v1/2021.naacl-demos.6.

[116] Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261. Available: http://arxiv.org/abs/1903.12261

[117] Miller, J. et al. (2020). The effect of natural distribution shift on question answering models. International Conference on Machine Learning (ICML), 119, 6905-6916. PMLR. Available: http://proceedings.mlr.press/v119/miller20a.html

[118] Geirhos, R. et al. (2020). Shortcut learning in deep neural networks. Nature Machine Intelligence, 2(11), 665-673. doi: 10.1038/s42256-020-00257-z.

[119] Oakden-Rayner, L. et al. (2020). Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. ACM conference on health, inference, and learning, 151-159. doi: 10.1145/3368555.3384468.

[120] Sohoni, N. et al. (2020). No Subclass Left Behind: Fine-Grained Robustness in Coarse-
Grained Classification Problems. Advances in Neural Information Processing Systems, 33, 19339-19352. Available: https://papers.nips.cc/paper/2020/file/e0688d13958a19e087e123148555e4b4-Paper.pdf

[121] Belinkov, Y. and Bisk, Y. (2017). Synthetic and natural noise both break neural machine translation. arXiv preprint arXiv:1711.02173. Available: http://arxiv.org/abs/1711.02173

[122] McCoy, T. et al. (2019). Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. Annual Meeting of the Association for Computational Linguistics, 3428-3448. doi: 10.18653/v1/P19-1334.

[123] Lipton, Z. C. and Steinhardt, J. (2018). Troubling trends in machine learning scholarship. arXiv preprint arXiv:1807.03341. Available: http://arxiv.org/abs/1807.03341

[124] Ethayarajh, K. and Jurafsky, D. (2020). Utility is in the Eye of the User: A Critique of NLP Leaderboards. Conference on Empirical Methods in Natural Language Processing (EMNLP), 4846–4853. doi: 10.18653/v1/2020.emnlp-main.393.

[125] Simmons, J. P. et al. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. Psychological science, 22(11), 1359-1366. doi: 10.1177/0956797611417632.

[126] Reimers, N. and Gurevych, I. (2017). Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. Conference on Empirical Methods in Natural Language Processing (EMNLP), 338–348. doi: 10.18653/v1/D17-1035.

[127] Melis, G. et al. (2017). On the state of the art of evaluation in neural language models. arXiv preprint arXiv:1707.05589. Available: http://arxiv.org/abs/1707.05589

[128] Zoph, B. and Le, Q. V. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578. Available: http://arxiv.org/abs/1611.01578

[129] Raff, E. (2019). A step toward quantifying independently reproducible machine learning research. Advances in Neural Information Processing Systems, 32, 5485-5495. Available: https://proceedings.neurips.cc/paper/2019/file/c429429bf1f2af051f2021dc92a8ebea-Paper.pdf

[130] Card, D. et al. (2020). With Little Power Comes Great Responsibility. Conference on Empirical Methods in Natural Language Processing (EMNLP), 9263–9274. doi: 10.18653/v1/2020.emnlp-main.745.

[131] Button, K. S. et al. (2013). Power failure: why small sample size undermines the
reliability of neuroscience. Nature reviews neuroscience, 14(5), 365-376. doi: 10.1038/nrn3475.

[132] Zheng, S. et al. (2016). Improving the robustness of deep neural networks via stability training. IEEE conference on computer vision and pattern recognition (CVPR), 4480-4488. doi: 10.1109/CVPR.2016.485.

[133] Dodge, S. and Karam, L. (2017). A study and comparison of human and deep learning recognition performance under visual distortions. International conference on computer communication and networks (ICCCN), 1-7. IEEE. doi: 10.1109/ICCCN.2017.8038465

[134] Buolamwini, J. and Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. Conference on fairness, accountability and transparency, 81, 77-91. PMLR. Available: http://proceedings.mlr.press/v81/buolamwini18a.html

[135] Dodge, J. et al. (2019). Show Your Work: Improved Reporting of Experimental Results. 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2185-2194. doi: 10.18653/v1/D19-1224.

[136] Bouthillier, X. et al. (2021). Accounting for variance in machine learning benchmarks. Machine Learning and Systems, 3, 747-769. Available: https://proceedings.mlsys.org/paper/2021/file/cfecdb276f634854f3ef915e2e980c31-Paper.pdf

[137] Thompson, W. H. et al. (2020). Meta-research: Dataset decay and the problem of sequential analyses on open datasets. Elife, 9, e53498. doi: 10.7554/eLife.53498.

[138] He, K. et al. (2016). Deep residual learning for image recognition. IEEE conference on computer vision and pattern recognition (CVPR), 770-778. doi: 10.1109/CVPR.2016.90.

[139] DeVries, T. and Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552. Available: http://arxiv.org/abs/1708.04552

[140] Yun, S. et al. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. IEEE/CVF International Conference on Computer Vision (ICCV), 6023-6032. doi: 10.1109/ICCV.2019.00612.

[141] Zhang, H. et al. (2017). Mixup: Beyond empirical risk minimization. arXiv preprint
[142] Hendrycks, D. et al. (2019). Augmix: A simple data processing method to improve robustness and uncertainty. arXiv preprint arXiv:1912.02781. Available: http://arxiv.org/abs/1912.02781

[143] Cubuk, E. D. et al. (2019). Autoaugment: Learning augmentation strategies from data. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 113-123. doi: 10.1109/CVPR.2019.00020.

[144] Sagawa, S. et al. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. arXiv preprint arXiv:1911.08731. Available: http://arxiv.org/abs/1911.08731

[145] Zech, J. R. et al. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. PLoS medicine, 15(11), e1002683. doi: 10.1371/journal.pmed.1002683.

[146] Foster, E. D. and Deardorff, A. (2017). Open science framework (OSF). Journal of the Medical Library Association: JMLA, 105(2), 203. doi: 10.5195/jmla.2017.88.