A large-scale evaluation framework for EEG deep learning architectures

Felix A. Heilmeyer
Translational Neurotechnology Lab
University Medical Center Freiburg
Freiburg, Germany
science@fehe.eu

Robin T. Schirrmeister
Translational Neurotechnology Lab
University Medical Center Freiburg
Freiburg, Germany
robin.schirrmeister@uniklinik-freiburg.de

Lukas D. J. Fiederer
Translational Neurotechnology Lab
University Medical Center Freiburg
Freiburg, Germany
lukas.fiederer@uniklinik-freiburg.de

Martin Völker
Translational Neurotechnology Lab
University Medical Center Freiburg
Freiburg, Germany
martin.voelker@uniklinik-freiburg.de

Joos Behncke
Translational Neurotechnology Lab
University Medical Center Freiburg
Freiburg, Germany
joos.behncke@uniklinik-freiburg.de

Tonio Ball
Translational Neurotechnology Lab
University Medical Center Freiburg
Freiburg, Germany
tonio.ball@uniklinik-freiburg.de

Abstract—**EEG is the most common signal source for noninvasive BCI applications. For such applications, the EEG signal needs to be decoded and translated into appropriate actions. A recently emerging EEG decoding approach is deep learning with Convolutional or Recurrent Neural Networks (CNNs, RNNs) with many different architectures already published. Here we present a novel framework for the large-scale evaluation of different deep-learning architectures on different EEG datasets. This framework comprises (i) a collection of EEG datasets currently including 100 examples (recording sessions) from six different classification problems, (ii) a collection of different EEG decoding algorithms, and (iii) a wrapper linking the decoders to the data as well as handling structured documentation of all settings and (hyper-) parameters and statistics, designed to ensure transparency and reproducibility. As an applications example we used our framework by comparing three publicly available CNN architectures: the Braindecode Deep4 ConvNet, Braindecode Shallow ConvNet, and two versions of EEGNet. We also show how our framework can be used to study similarities and differences in the performance of different decoding methods across tasks. We argue that the deep learning EEG framework as described here could help to tap the full potential of deep learning for BCI applications.**

Index Terms—**EEG, BCI, Deep Learning, Convolutional Neural Networks, Braindecode, EEGNet, FBCSP, Performance Comparison**

I. Introduction

EEG is the most common signal source for noninvasive BCI applications. For such applications to work reliably, the EEG signal needs to be decoded with high accuracy and translated into appropriate actions. To this purpose a large and growing variety of decoding methods is being used. A recently emerging EEG decoding approach is deep learning with Convolutional or Recurrent Neural Networks (CNNs, RNNs). Deep learning has already revolutionized other areas and excels at decoding information form raw data, e.g., in image recognition [1] and natural language processing [2]. Thus, currently many BCI researchers are starting to investigate the potential usefulness of deep learning techniques using a wide range of different network architectures and applying them to a wide range of EEG datasets [3]–[7].

Faced with this large variety in architectures and applications, choosing a network architecture for new BCI tasks is not trivial for various reasons. Although most of the published studies evaluated the performance of their deep learning architectures against some comparison algorithm, in most of the cases these comparisons were either against traditional, non-deep-learning decoding methods, or involved different versions of the newly introduced network architecture. Studies evaluating against other, already published CNN- or RNN-based analyses are less common. Moreover, the number of different datasets used in these evaluations is often small and may not reflect the wide range of EEG decoding problems of different difficulty [6], [8], [9].

Therefore, a framework for the systematic evaluation of deep learning for EEG which addresses these challenges is desirable. As a first step in this direction, we developed a framework to compare deep learning architectures on a large set of EEG examples (>100 decoding problems) in a comprehensively documented and reproducible manner. Applying this framework, here we report on three publicly available CNN architectures.

In this paper we describe the framework and provide our rationale for design choices regarding the challenges describes above. Secondly, we present the results of our comparison and make a recommendation on which of the network(s) included in the comparison to use for best performance. At last we discuss how the present framework could be further extended and improved.

TABLE I
DATASET SUMMARY

Name (Acronym)	# Classes	Task Type	# Subjects	Trials per Subject	Class balance
High-Gamma Dataset (Motor)	4	Motor task	20	1000	balanced
KUKA Pouring Observation (KPO)	2	Error observation	5	720-800	balanced
Robot-Grasping Observation (RGO)	2	Error observation	12	720-800	balanced
Error-Related Negativity (ERN)	2	Eriksen flanker task	31	1000	1/2 up to 1/15
Semantic Categories	3	Speech imagery	16	750	balanced
Real vs. Pseudo Words	2	Speech imagery	16	1000	3/1
Total			100		

II. METHODS

Our framework is build upon three components: (i) a collection of EEG data, (ii) the decoding methods embedded in the Braindecode toolbox, and (iii) a wrapper that enables running large-scale decoding experiments in an easily reproducible fashion. This section contains a detailed description of these three components.

A. EEG Data

The performance evaluation was done on a range of datasets representing a spectrum of common BCI tasks including motor tasks, speech imagery, and error processing. We choose to include different tasks to ensure that success or failure of the compared methods is not limited to a specific decoding domain. The difficulty of the included decoding problems ranges from data which can be decoded by all methods with high accuracy to data which is almost impossible to decode with currently available methods. One might argue that data too difficult to decode for all employed methods does not contribute to the comparison. We still included this data for the following reason: Assume we only included easy to decode data, on which all methods already achieve high decoding accuracies. In this scenario, if repeating the evaluation in the future with a potentially better performing method, that method could not show its full potential. In contrast, by including difficult data, on which current methods only achieve medium accuracies, we leave room for future methods to show their superiority. This allows us to subsequently expand this comparison to new emerging decoding methods.

All included datasets were acquired at the Translational Neurotechnology Lab, University of Freiburg, and recorded with an EEG cap with 128 gel-filled electrodes. During recording subjects were presented with different stimuli or had to perform specific tasks. The following paragraphs contain a short description of the six paradigms included in this study. For a detailed description of the data acquisition and experimental setups please refer to the respective cited original publications. Together, the datasets amounted to 100 different decoding task examples. Table I gives an overview of the included datasets.

Our first dataset was initially published in [6] and named "High-Gamma Dataset" (motor) as the recording setup was optimized to capture movement related frequencies in the high gamma range. For recording, subjects were instructed via visual stimuli to hold still, tap the fingers of either left hand, right hand, or flex the toes of both feet. The decoding task evaluated in this study is to classify which of those four instructions was executed at each trial. Recording data from stimulus onset to 4 s after onset was used for decoding.

Our second dataset dubbed "Error-Related Negativity" (ern) used a variant of the Eriksen flanker task [10]. This involves reacting as fast as possible to a visual stimulus by pressing a button with either the left or right index finger. When subjects reacted to the stimulus with pressing the correct button, they were rewarded with points; When they pressed the wrong button, they were penalized by losing points. The according decoding problem is to classify for each trial if the subject was successful or failed, that is pressing the correct or the wrong button. The data used for decoding in this study was the EEG recorded from 0.5 s before to 1.5 s after a button was pressed. A detailed description can be found in [11].

In the KUKA Pouring Observation paradigm (kpo) subjects were watching a video of a robotic arm pouring liquid from a bottle into a glass. In each trial the robot either succeeded or spilled the liquid. On this data the task for the decoder is to classify whether the subject was watching a video of a successful or unsuccessful attempt.

The Robot-Grasping Observation paradigm (rgo) consisted of subjects watching a video of a robot approaching, grabbing and then lifting a ball from the floor. Similar to the previous experiment, in each task the robot was either failing or succeeding; And the decoding task is to classify whether the subject was watching failure or success.

In both observation paradigms the error occurred in an interval from 2.5 s to 5 s after stimulus start. Consequently, data recorded in this interval was used for decoding. Publications featuring detailed descriptions of these two datasets are [12] [13].

The last dataset used in this study concerned semantic processing. It was first published in [14]. The recording setup matches the setup used for recording the previous datasets. The subjects were presented with a word on a computer screen for 500 ms and instructed to repeat the word silently for three seconds following the stimulus. Data recorded in these three seconds was later utilized for decoding. The presented words were 84 concrete nouns of three semantic categories: food, animals, and tools. Additionally, an equal number of pseudowords was included. These pseudowords were constructed to look and feel like real words but carry no meaning [15]. We created two separate decoding tasks from this dataset, one with

TABLE II
CLASSIFIER ARCHITECTURES

	Deep	Shallow	EEGNet v1	EEGNet v2
Layer 1	25xConv2D (10, 1), Stride 1,1 25xConv2D (1, C), Stride 1,1 BatchNorm Activation (ELU) MaxPool (3, 1), Stride 3,1	40xConv2D (25, 1), St1,1 40xConv2D (1, C), St1,1 BatchNorm Activation (Square) MeanPool (75, 1), St15,1 Activation (Log) Dropout (0.5)	16xConv1D (C, 1), St1,1 BatchNorm Activation (ELU) Transpose to 16,T Dropout (0.25)	8xConv2D (1,64), St1,1 BatchNorm 16xDepthwiseConv2D (C,1), St1,1 BatchNorm Activation (ELU) AveragePool2D (1,4), Stride 1,1 Dropout (0.25)
Layer 2	Dropout (0.5) 50xConv2D (10, 1), Stride 1,1 BatchNorm Activation (ELU) MaxPool (3, 1), Stride 3,1	Dense Softmax Classification	4xConv2D (2,32), St1,1 BatchNorm Activation (ELU) MaxPool (2,4), St1,1 Dropout (0.25)	16xSeparableConv2D (1,6), Str1, 1 BatchNorm Activation (ELU) AveragePool2D (1,8), St1,1 Dropout (0.25)
Layer 3	Dropout (0.5) 100xConv2D (10, 1), Stride 1,1 BatchNorm Activation (ELU) MaxPool (3, 1), Stride 3,1		4xConv2D (8,4), St1,1 BatchNorm Activation (ELU) MaxPool (2,4), St1,1 Dropout (0.25)	Dense Softmax Classification
Layer 4	Dropout (0.5) 200xConv2D (10, 1), Stride 1,1 BatchNorm Activation (ELU) MaxPool (3, 1), Stride 3,1		Dense Softmax Classification	
Layer 5	Dense Softmax Classification			

three and one with two classes, respectively, by (i) labeling trials with one of the three semantic categories (omitting the pseudowords) and by (ii) distinguishing real vs pseudowords (semantic and pseudovsreal, respectively).

The following paragraphs outline how the data was preprocessed before decoding and why this specific preprocessing approach was chosen.

The preprocessing was the same for all methods and all datasets. Most likely, decoding performance would have profited from individualized preprocessing. Nevertheless, we choose to use a uniform preprocessing to ensure that observed performance differences were the effect of the decoding method alone. First, the data was downsampled to 250 Hz and bandpass filtered with a 0.5–120 Hz filter. This preserves the main neurophysiologically important frequency bands usually considered in EEG recordings but also reduces the amount of data to allow reasonable training duration, even with deep CNNs.

At this point the last 20% of every subject's recording were split and put aside for final testing. The cleaning procedures described in the following were only applied to the training data as it is customary in developing many machine learning applications. Thereby, the testing results reflect actual decoding performance when the trained classifier is applied to new data without cleaning. We restricted cleaning to the recording artifacts employing the following algorithm: First, following our procedure as described in [6], all channels in which more than 20% of the samples were over 800 µV were marked as broken and removed. Then trials were cut and any trial which still contained samples over 800 µV was removed. This simple cleaning mechanism thus removed large-amplitude artifacts that would likely disturb the training process.

B. Decoding methods

In this section we will shortly introduce the decoding methods compared in this study. For an overview of all included architectures refer to Table II. To avoid errors we used the implementations of the original authors for all decoding methods. For implementation details refer to the respective publications.

Firstly, we used two CNN architectures that are part of the Braindecode open source toolbox for EEG decoding released last year by our lab and published in [6]: The Braindecode Deep4 ConvNet and Braindecode Shallow ConvNet, hereafter referred to as Deep4 Network and Shallow Network. The Deep4 Network features four convolution-max-pooling blocks, using batch normalization and dropout, followed by a dense softmax classification layer. The first block is split, first performing a temporal convolution then a spatial convolution over all channels followed by the max pooling. The other three blocks are standard convolution-max-pooling blocks. All layers use exponential linear units (ELUs) as nonlinearities.

The Shallow Network architecture also features a temporal then a spatial convolution layer, followed by a squaring nonlinearity, a mean-pooling layer and a dense classification layer. This architecture has many similarities to the FBCSP method. For further details refer to [6].

The third decoding method in this comparison was a CNN called EEGNet [9] designed for compactness (few trainable parameters). It consists of three convolutional layers followed by a softmax regression layer. The first convolutional layer only convolutes spatially; Following layers also convolute temporal and include max pooling. Additionally, every layer uses batch normalization and dropout.

Shortly before the submission of this paper the authors of

EEGNet published an updated version of EEGNet with a fundamentally changed architecture [8], which we will refer to as EEGNetv2 in this paper. The new EEGNetv2 architecture uses first temporal then spatial convolutions in its first layer. Hence the initial layers are now more similar to the Braindecode networks compared to the original EEGNet. In contrast to our the Braindecode networks EEGNetv2 uses a depthwise convolution with a depth multiplier of two, therefore learning two separate spatial filters for each location. Additionally, the second layer of EEGNetv2 uses separable instead of standard convolutions. Lastly, EEGNetv2 uses average pooling, whereas the original EEGNet uses max pooling. For an overview of the architecture differences refer to Table II.

C. Comparison wrapper, statistics and correlation analysis

The last component of our framework is a wrapper that enables large scale comparison experiments. It was designed with a focus on transparency and reproducibility by putting all the information needed to rerun a comparison in a comprehensive configuration package. Additionally, the wrapper automatically generates most of the statistics and plots contained in this paper. This section contains a description of the classifier training and testing setup, i.e., the hyperparameter setup as well as the statistics used in the analysis of the results.

For setting the training hyperparameters, we adopted the settings as proposed in the original publications on each dataset as cited above. All models were trained by optimizing the categorical cross-entropy loss using the Adam optimizer from [16]. All training setups including the training duration match the original publications of the respective architectures.

To test for significance, a random permutation test was used. This was done by randomly permuting the assignment of test labels to test trials and calculating the resulting accuracy on these randomly permuted labels. By repeating this process 10^6 times a null distribution was created. Comparing the actual decoding accuracy to the distribution allowed an estimation of how likely it is that a given decoding result was achieved at random, i.e., the significance of the decoding. Following this process, for each recording the significance of the decoding for all four included classifiers was tested. Then, recordings where the p-value exceeded 5% for all four classifiers were excluded from all further analysis.

As described above the data includes datasets which are difficult to decode. These datasets would add noise to the comparison in cases where all methods yield equally low around-chance-level results. Therefore, only recordings on which at least one of the tested methods achieved a significantly above chance decoding accuracy were further analyzed.

All decoding accuracies mentioned in this paper were calculated as mean class accuracies, by first calculating the accuracy separately for each class and then taking the mean of these accuracies across classes. Thereby the chance level is always at 1/number of classes irrespective of the distribution of examples across classes in the testing data.

The varying difference in task difficulty makes for a broad spread in the distribution of absolute decoding accuracies. To compare these across the four different classifiers, besides the absolute accuracies, we also calculated normalized accuracies by dividing the accuracies obtained with each classifier by the mean across all four different classifiers that we evaluate. Significance of the differences between decoding accuracies was calculated with a two-sided sign test, testing the hypothesis that the difference between to result distributions does not have a zero median.

In addition to comparing the methods' performance as such, we also assessed the correlations of the predictions of the different decoders, both across decoding task examples and on a trial-by-trial basis. One motivation for this analysis was to see whether different methods potentially succeed and fail in different trials; If so, such differences could give hints about the functional properties of different methods and ensembles and combining multiple of the investigated classifiers might allow further decoding accuracy improvements. Therefore, for each method pair, both the prediction dissociation and overlap was computed, i.e., the percentage of trials where both methods either succeed or fail to predict the correct class, as well as the percentage of trials where one method was correct and the other was not.
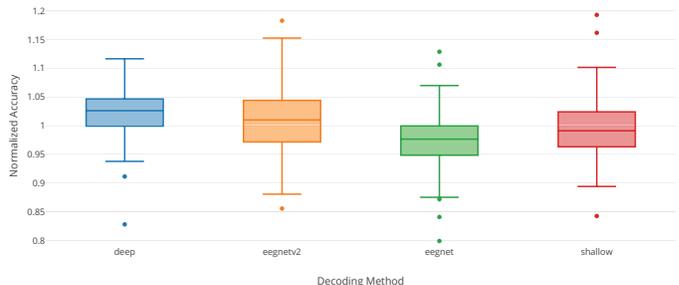

Fig. 1. Decoding results normalized to the average performance across classifiers. The Deep4 Network performed better than EEGNet and the Shallow Network, the latter two being almost equal. The new EEGNetv2 architecture improved significantly, now on par with the Deep4 Network.

TABLE III
DECODING RESULTS

	Mean accuracy	Mean normalized accuracy
Deep Network	70.08% ± 20.92%	1.00 ± 0.05
EEGNetv2	70.00% ±18.86%	1.02 ± 0.08
EEGNet	67.71% ± 19.04%	0.98 ± 0.06
Shallow Network	67.71% ±19.04%	0.99 ± 0.06

III. RESULTS

A. Performance comparison

In 75 out of the 100 decoding task examples, at least one classifier achieved a significantly above chance accuracy (p < 0.05). Only results on these 75 examples are included
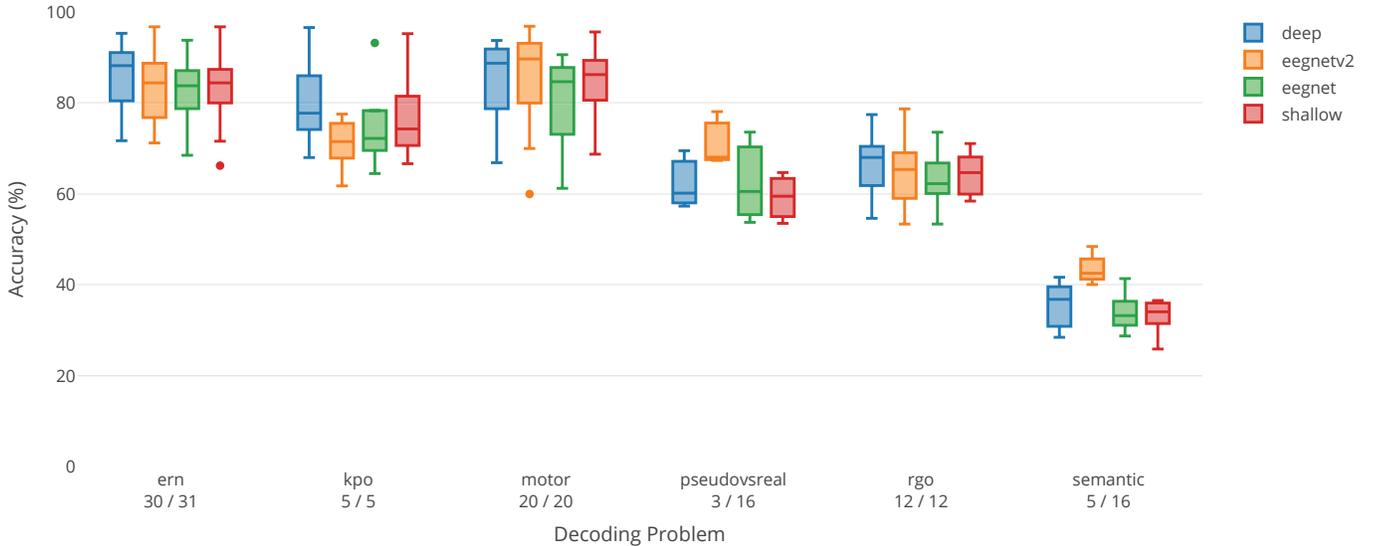
Fig. 2. Class mean decoding accuracy per decoding task and decoding method. The overall performance ranking seen in Fig. 1 applied to most individual decoding tasks as well. Note that the box-plots only refer to the examples where at least one of the classifiers yielded a significant result, the number of which is indicated on the bottom together with the total number of examples. For instance in the motor dataset, this included all 20 examples, while in the pseudovsreal semantic decoding problem only three subject had successful decodings with at least one method, illustrating the wide range of difficulty covered by our collection of datasets.

in the following analyses. Fig. 1 and Table III give an overview of the overall performance of the four different classifiers compared in the present study. The Deep4 Network performed significantly better than the Shallow Network and first generation EEGNet. The new architecture of EEGNetv2 significantly improved the decoding accuracy compared to its first generation architecture, with a performance of EEGNetv2 not significantly different to that of the Deep4 Network.

This overall performance ranking is also reflected in the results split according to the six different decoding tasks shown in Fig. 2. Fig. 2 also illustrates the different levels of difficulty of the tasks; in ERN, KPO, motor and RGO, in most of the individual examples significant results were achieved at least by one of the tested methods, in contrast to the much harder semantic tasks.

As also illustrated by Fig. 3, the results of the performance comparison can be summarized as follows: Among the CNNs the Deep4 Network yielded significantly better performance than first generation EEGNet, whereas the Shallow Network's performance settled in a middle ground between Deep4 Network and EEGNet: The difference between the first and neither of the latter two reached significance. The new architecture of EEGNetv2 significantly improved the decoding accuracy, which is now on par with Deep4 Network's performance.

B. Prediction correlation

The results of the prediction correlation analysis comparing the four different decoders is shown in Fig. 3. Note that the scatter plots also include results on the 25 examples excluded from analysis as gray dots. Fig. 3 shows that all CNNs disagreed in their predictions on about 15-20% of the trials. Furthermore, the distribution of correct predictions

was relatively balanced. This indicates that ensembles could potentially outperform single CNNs.

IV. CONCLUSIONS

Here we have presented a novel framework for the large-scale evaluation of different deep-learning architectures on different EEG datasets. This framework comprises (i) a collection of EEG datasets, (ii) a collection of different EEG decoding algorithms already published by our lab in an open-source toolbox, and (iii) a wrapper linking the decoders to the data as well as handling structured documentation of all settings and (hyper-) parameters and statistics, designed to ensure transparency and reproducibility. As already done for component (ii), we strive to make the other two components publicly available to the BCI community in the near future as well.

As an application example of our framework, we have evaluated four CNN architectures. Our comparison also showed that among the CNN architectures evaluated, the Braindecode Deep4 Network and EEGNetv2 performed best. The new EEGNetv2 not only produces similar results compared to our Deep4 Network, but also is architecturally closer to our networks than to the original EEGNet design. On the other hand, the EEGNetv2 architecture is more compact allowing faster training than the Deep4 Network while simultaneously achieving equally high accuracy. We suspect that this is may partially due to the use of separable and depthwise convolutions. We will therefore investigate the possibility of improving the performance of the Braindecode Deep4 Network with these convolution variants. Furthermore, it would also be interesting to investigate potential advantages of deeper networks especially for large EEG datasets.
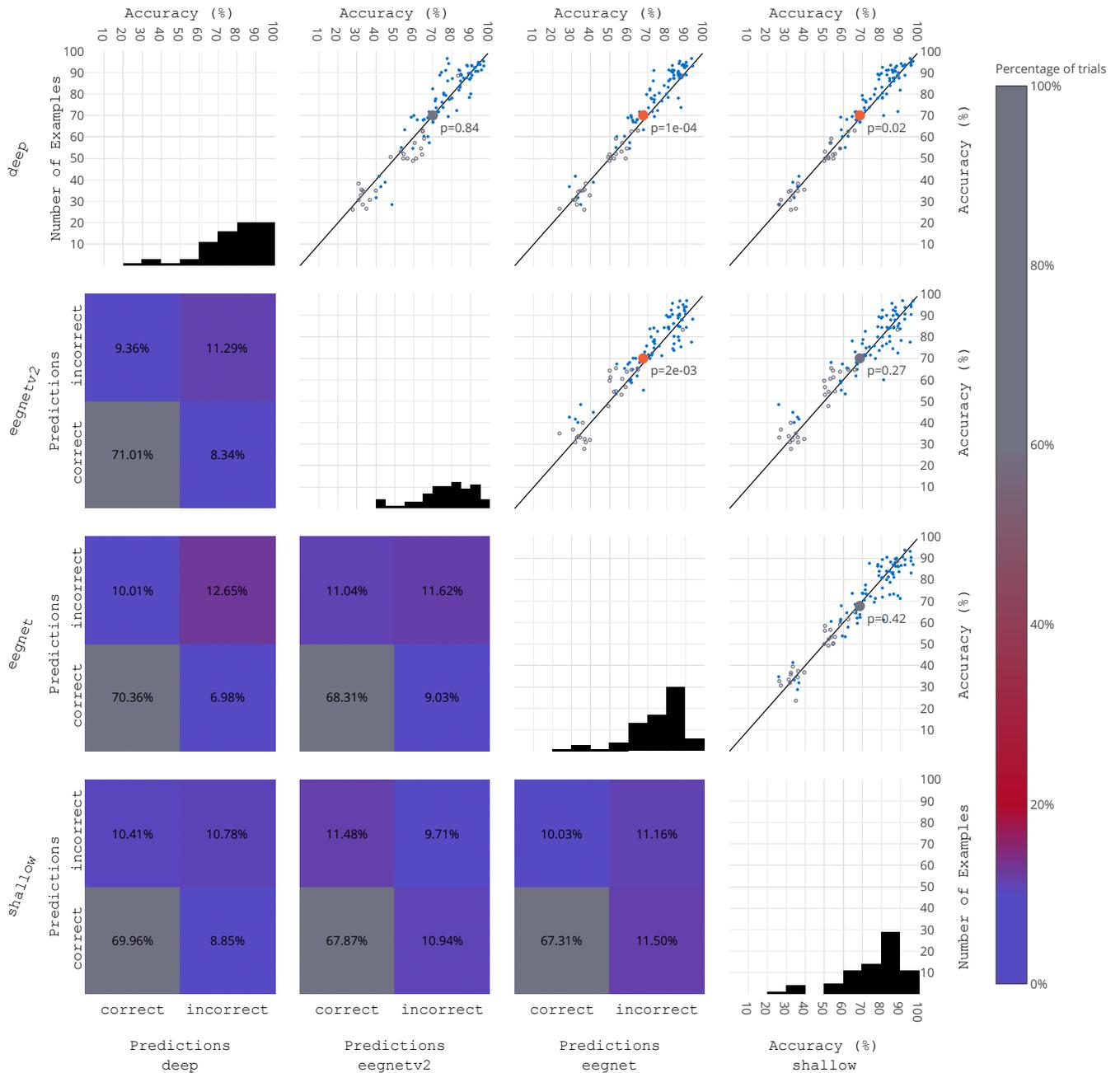
Fig. 3. Performance correlations. Rows and columns are decoding methods. The subplots in the diagonal show the histograms of the decoding accuracies of the individual methods. Offdigonal plots show comparisons of method pairs. The upper-right plots are pairwise comparisons of the decoding performance per examples (subjects). Blue dots indicate examples with at least one significant decoding result, open gray circles indicate recordings where no method achieved a significant result. Large dots indicate the mean accuracy over significant examples. The mean was plotted in orange if the overall difference between two methods was also significant (p < 0.05), else in gray, with the corresponding p-value printed next to it. The lower-left plots show color coded matrices of the correlation of correct and incorrect predictions. All X axes of the pairwise plots correspond to the method of the respective column, Y axes to the method of the corresponding row.

Simultaneously, there is an even larger number of other CNN and also RNN architectures that have been proposed for EEG decoding and that are not yet available within our (or any other) framework for large-scale evaluation of such methods. In the future, we will add more deep learning EEG-decoding methods and make them available within our framework. We believe that a comprehensive collection of different decoding models that are all compatible with a large collection of EEG test data, as initiated here, could become very helpful for identifying architectures that meet the requirements of specific

would hope that methods developed by other researchers in the field would become available in a compatible form as well. We believe that a comprehensive collection of different decoding models that are all compatible with a large collection of EEG test data, as initiated here, could become very helpful for identifying architectures that meet the requirements of specific

research and application scenarios.

In parallel to extending the collection of decoding models, as another future aim we plan to extend our collection of EEG datasets. Possible extensions would be, for example, datasets on the widely used P300 speller paradigms, datasets on "passive BCI" decoding problems such as work load or attention decoding, sleep staging, etc., as well as datasets reflecting a broader range of EEG acquisition techniques and conditions, such as dry EEG or mobile recordings. With a growing data base, a framework as proposed here could also become a useful tool for automatic hyperparameter optimization and architecture search, by providing the large amounts of data typically required by such techniques.

Furthermore, our framework could possibly be developed into a EEG decoding challenge or even an ongoing public benchmark in the likes of the Stanford Question Answering Dataset (SQuAD) [17]. Challenges like SQuAD, the ImageNet Competition or the many competitions hosted on the Kaggle platform have benefited progress in many areas where deep learning is applied but are still scarce in the BCI area. In the present study we choose "out of the box" performance evaluation by setting the hyperparameters according to the original publications of the architectures investigated, as an appropriate metric for initial architecture comparison. This is most likely not optimal as the networks may have performed better when using approaches for systematic hyperparameter optimization. Therefore, comparison of such optimized versions of the different proposed architectures would be an interesting future research topic. However, it is unlikely that a single research institution would be able to keep up with all new/updated architecture releases and perform such optimized comparisons. Therefore, a public benchmark would be helpful to alleviate this problem by distributing the workload to all participants while providing comparable results, enabling a comprehensive overview of the state of the art decoding methods' performance. Accordingly, our next step will be to establish such a benchmark utilizing our extensive EEG data collection.

The possibility for evaluation of different deep learning techniques that is large-scale both with respect to the number of network models and datasets included, could be helpful for an informed initial choice of a suitable deep learning architecture also for novel EEG decoding problems. Such an informed initial choice would be especially important when considering deep learning for online BCIs, as in [7], where most of the data is acquired after having fixed the decoder architecture. Thus, in summary, the deep learning EEG framework as described here could help to tap the full potential of deep learning for BCI applications.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[2] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *arXiv.org*, Aug. 2014.

[3] P. Bashivan, I. Rish, M. Yeasin, and N. Codella, "Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks," *arXiv.org*, Nov. 2015.

[4] Z. Tang, C. Li, and S. Sun, "Single-trial EEG classification of motor imagery using deep convolutional neural networks," *Optik - International Journal for Light and Electron Optics*, vol. 130, pp. 11–18, Feb. 2017.

[5] M. Lee, S.-K. Yeom, B. Baird, O. Gosseries, J. O. Nieminen, G. Tononi, and S.-W. Lee, "Spatio-temporal analysis of EEG signal during consciousness using convolutional neural network," in *2018 6th International Conference on Brain and Computer Interface (BCI)*. IEEE, 2018, pp. 1–3.

[6] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization." *Human brain mapping*, vol. 38, no. 11, pp. 5391–5420, Nov. 2017.

[7] F. Burget, L. D. J. Fiederer, D. Kuhner, M. Völker, J. Aldinger, R. T. Schirrmeister, C. Do, J. Boedecker, B. Nebel, T. Ball, and W. Burgard, "Acting thoughts: Towards a mobile robotic service assistant for users with limited communication skills," in *2017 European Conference on Mobile Robots (ECMR)*, Sep. 2017.

[8] V. Lawhern, A. Solon, N. Waytowich, S. M. Gordon, C. Hung, and B. J. Lance, "EEGNet: a Compact Convolutional Neural Network for EEG-Based Brain-Computer Interfaces." *Journal of Neural Engineering*, Jun. 2018.

[9] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces," *arXiv.org*, Nov. 2016.

[10] C. W. Eriksen and B. A. Eriksen, "Target redundancy in visual search: Do repetitions of the target within thedisplay impair processing?" *Perception & Psychophysics*, vol. 26, no. 3, pp. 195–205, May 1979.

[11] M. Völker, L. D. Fiederer, S. Berberich, J. Hammer, J. Behncke, P. Kršek *et al.*, "The dynamics of error processing in the human brain as reflected by high-gamma activity in noninvasive and intracranial EEG," *NeuroImage*, 2018.

[12] J. Behncke, R. T. Schirrmeister, W. Burgard, and T. Ball, "The signature of robot action success in EEG signals of a human observer: Decoding and visualization using deep convolutional neural networks," in *2018 6th International Conference on Brain-Computer Interface (BCI)*. IEEE, 2018.

[13] D. Welke, J. Behncke, M. Hader, R. T. Schirrmeister, A. Schönau, B. Eßmann, O. Müller, W. Burgard, and T. Ball, "Brain Responses During Robot-Error Observation," *arXiv.org*, Aug. 2017.

[14] V. Rau, "EEG Correlates of Inner Speech," Bachelor's Thesis, University of Freiburg, DOI 10.6094/UNIFR/15003, 2015.

[15] G. Blanken, R. Döppler, M. Bautz, and K. J. Schlenck, *Wortproduktionsprüfung*. NAT-Verlag, 1999.

[16] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv.org*, Dec. 2014.

[17] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100, 000+ questions for machine comprehension of text," *CoRR*, vol. abs/1606.05250, 2016. [Online]. Available: http://arxiv.org/abs/1606.05250