Amplified Early Stopping Bias: Overestimated Performance with Deep Learning

Nona Rajabi
Department of Intelligent Systems
KTH Royal Institute of Technology
Stockholm, Sweden
nonar@kth.se

Antônio H. Ribeiro
Department of Information Technology
Uppsala University
Uppsala, Sweden
antonio.horta.ribeiro@it.uu.se

Miguel Vasco
Department of Intelligent Systems
KTH Royal Institute of Technology
Stockholm, Sweden
miguelsv@kth.se

Danica Kragic
Department of Intelligent Systems
KTH Royal Institute of Technology
Stockholm, Sweden
dani@kth.se

Abstract

Cross-validation is commonly used to estimate machine learning model performance on new samples. However, using it for both hyperparameter selection and error estimation can lead to overestimating model performance, especially with extensive hyperparameter searches that overly tailor models to validation data. We demonstrate that deep learning further amplifies this bias, with even minor model adjustments causing significant overestimation. Our extensive experiments on simulated and real data focus on the bias from early stopping during cross-validation. We find that overestimation intensifies with network depth and is especially severe in small datasets, which are common in physiological signal processing applications. Selecting the early stopping point during cross-validation can result in ROC-AUC estimates exceeding 90% on random data, and this effect persists across various sample sizes, architectures, and network sizes. All codes are publicly available at https://github.com/NonaRjb/DeepOverestimation.git.

1 Introduction

The success of deep learning (DL) in data-rich fields like text and image processing [1, 2, 3] has extended its application to areas with limited data, often achieving significant success. Specifically, for physiological signals, DL consistently outperforms other methods in competitions [4, 5], even with smaller-scale datasets [6, 7]. Small sample sizes are common in human studies due to the high data collection costs and the need for short sessions to avoid participant fatigue. The expenses for participant recruitment, experimental setup, and ensuring data quality are significant, especially in the early stages of research when the value of the data is uncertain. Hence, increasing the sample size is often infeasible, prompting deep neural networks to be trained on limited data, with cross-validation to validate performance. This paper explores how applying cross-validation to estimate the performance of a DL model requires extra care, as neglecting the independence of model and data decisions can lead to a huge overestimation of performance.

Previous research has emphasized the importance of using a held-out test set, in addition to training and validation sets, when evaluating shallow machine learning (ML) models [8, 9, 10]. Tuning a model's hyperparameters based on validation performance can lead to overestimating the performance, where the model becomes overly tailored to the validation data. In this paper, we show that in low-data

Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

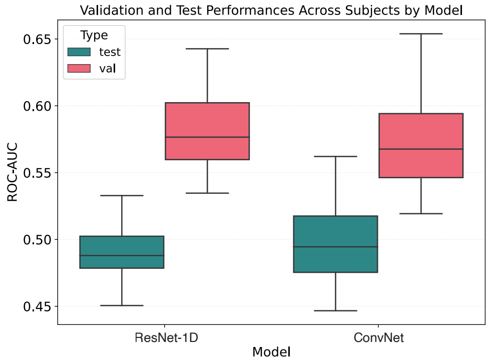


Figure 1: **Cross-validation and test performance** of ResNet-1D and ConvNet trained on EEG activity with random binary labels.

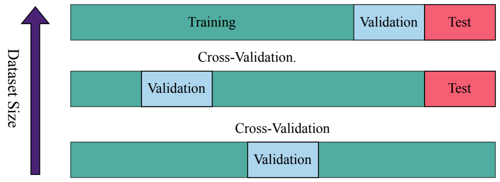


Figure 2: **Data splits for model selection and evaluation**: (top) large datasets typically use a single split for realistic performance estimation; (middle) for smaller datasets, which less accurately represent the underlying data distribution, cross-validation improves estimates; (bottom) with limited data, researchers may omit the test set, relying solely on cross-validation.

regimes, *merely choosing the early stopping point based on validation performance can significantly overestimate model performance*. An extensive ablation study confirms this consistent effect across different network architectures, sizes, optimization methods, and both synthetic and real data (Fig. 1). This highlights how DL can amplify biases, *with extreme cases yielding a ROC-AUC of 0.95 on completely random data*.

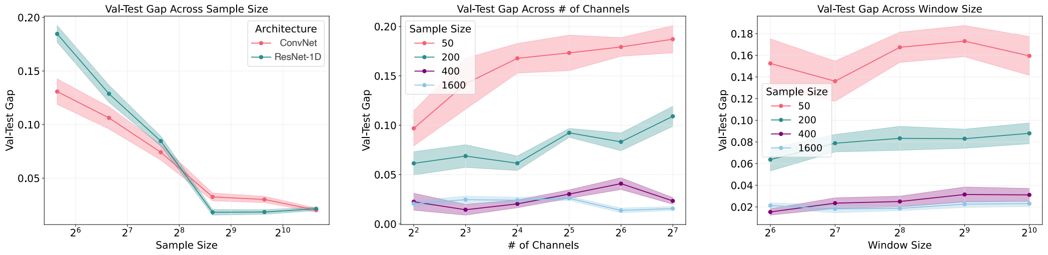
2 Setup

We evaluate the consistent effect of early stopping on the overestimation of the performance of DL models in the low-data regime. We focus on three fundamental components of a standard ML pipeline: the *data* used to train the model (sample size–the number of samples available for training and validation, feature size), the *network* used to instantiate the model (the architecture, the depth and the width of the network), and the *training* method employed (the optimizer). We summarize these parameters for each experiment in the Appendix (Table A1).

Model Selection and Evaluation. Data splitting is crucial for training models, tuning hyperparameters, and assessing performance. Typically, a dataset is divided into three sets: (1) training, (2) validation, and (3) test. The training and validation sets are used for model development, while the test set is reserved for final evaluation. Different splitting strategies for various dataset sizes are shown in Fig. 2. In this work, we show that excluding the test set in low-data scenarios leads to significant biases in performance estimates, even when performing very minor model adjustments.

Early Stopping. Early stopping is a regularization technique used in iterative ML to prevent overfitting. Iterative optimization updates model parameters to improve performance, but early stopping halts training before the model overfits the training data, thus preserving generalizability. Data is split into training and validation sets for this purpose, and the model's training ceases once the validation error starts increasing. Cross-validation can ensure evaluation robustness. We study the performance of neural networks that are early-stopped based on the minimum validation loss.

Experimental Setup and Evaluations Metrics. In each experiment, we train a neural network for up to 500 epochs, using early stopping based on minimal validation loss, and evaluate it on a held-out test set using the area under the receiver operating characteristic curve (ROC-AUC) as the metric. The ROC-AUC plots the true positive rate ($\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$) against the false positive rate ($\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$), with values above 0.5 suggesting better-than-chance outcomes. We investigate the gap between the cross-validation and test ROC-AUC of a neural network trained on random data. Formally, let $\mathcal{X}$ represent the dataset, with $\mathcal{X}_v$ as the validation set for early stopping, and $\mathcal{X}_t$ as the unseen test set. If $f : \mathcal{X} \to \mathcal{Y}$ denotes the trained model mapping samples $x \in \mathcal{X}$ to labels $y \in \mathcal{Y}$, we focus on the quantity $\frac{1}{K}\sum_{i=1}^{K}[\mathrm{AUC}(f_i(\mathcal{X}_{v_i})) - \mathrm{AUC}(f_i(\mathcal{X}_t))]$, where K is the number of validation sets in cross-validation and i is the i^{th} split. We call this metric the *validation-test gap*.



(a) Sample Size and Architectures (b) Channel Count and Sample Size (c) Window Size and Sample Size

Figure 3: **Effect of various parameters on the validation-test gap** for trained on synthetic data. Line plots show the performance averaged over different (a) channel counts and window sizes, (b) window sizes and architectures, and (c) channel counts and architectures. The shaded area indicates the standard error of the mean. The test performance remains close to 50% (random performance) in all the experiments.

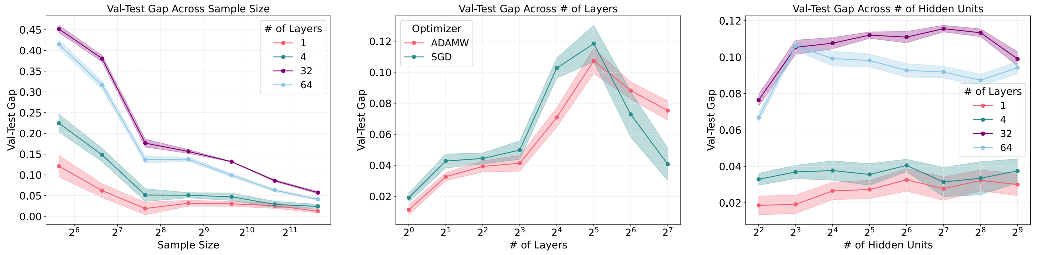
3 Experiments

We organized our experiments into three groups: (i) real physiological data, (ii) synthetic time series data, and (iii) Gaussian random vectors. The first experiment showcases the effect of overestimated performance in a real-world scenario. The remaining experiments provide greater control over data generation parameters: in (ii) we use time series data similar to the real case but vary the sample size, channel count, and window size; in (iii) we use non-time-series data with multi-layer perceptron networks, to simplify the setup and enhance interpretability.

Real Data. To demonstrate the validation-test gap in a real-world scenario, we employed electroencephalogram (EEG) data from 52 participants, recorded during an olfactory task. Signals were recorded from 64 scalp electrodes when participants underwent 140 trials (details in Section A.2). Some trials were excluded due to noise and artifacts. We selected 500 ms of *pre-stimulus baseline* activity. Also, the original labeling of the data was discarded and it was randomly and equally labeled as classes 0 and class 1, resulting in an expected classifier performance of 50%. We trained two convolutional neural networks (CNNs) on each participant's data: a shallow CNN based on the shallow ConvNet architecture [11] and a deep CNN inspired by the ResNet-1D architecture [12]. Nested cross-validation (details in Section A.3) was used to evaluate the cross-validation and test performance. The models were trained as explained in Section 2 and tested on the held-out test set. Figure 1 presents the cross-validation and test ROC-AUC of the two models across different participants. The data points used to create the box plots are the average ROC-AUC per participant, either from cross-validation or test sets. The plots reveal a clear median performance gap of approximately 8% between cross-validation and test results for both models.

Synthetic Data. Our second experiment was based on synthetic time series data. We utilized the signal generator function from Yeung et al. [13] to create baseline EEG-like activity (further details in Section A.4). The signals were labeled following the same process as the previous evaluation. Initially, 5000 samples of the signal were generated and held out as the test set. The results show that the validation-test gap decreases significantly with increasing the sample size (Fig. 3a). In addition, for very small sample sizes, increasing the channel count results in a gentle increase in the validation-test gap (Fig. 3b), although the effect is not observed by increasing the temporal window size (Fig. 3c).

Gaussian Random Vectors. We designed our third experiment using Gaussian random vectors. A multi-layer perceptron (MLP) was used instead of the CNN to explore the effect of network depth and width. Data samples were generated from a multivariate Gaussian distribution with a diagonal covariance matrix. Two diagonal values were set to 10, and the rest to 0.01, mimicking real-world scenarios where data often have a few large eigenvalues [14, 15, 16]. The labels were generated in the same manner as in the previous two experiments. Similar to the previous experiment, 5000 samples were generated at the beginning of the experiment to serve as the held-out test set. Figure 4a (and Suppl. Fig. A1) shows that with a small sample size, the cross-validation ROC-AUC can reach up



(a) Sample Size and # of Layers

(b) # of Layers and Optimizer

(c) # of Hidden Units and Layers

Figure 4: **Effect of various parameters on the validation-test gap** for an MLP trained on Gaussian random vectors. Line plots show the performance averaged over different (b) numbers of hidden units and (a, c) input feature sizes. The shaded area indicates the standard error of the mean. The test performance remains close to 50% (random performance) in all the experiments.

to 95% on random data (the purple graph). It highlights the significant impact of sample size, with the validation-test gap decreasing exponentially as sample size increases. This effect intensifies as the number of layers increases until it drops when the network is too deep, probably due to gradient vanishing. This trend occurs consistently, regardless of the network's width or the optimizer used (Fig. 4b). Lastly, Fig. 4c demonstrates that while network width has little impact on the validation-test gap, deeper networks consistently show a larger gap.

4 Related Work

Deep Learning on Limited Data. DL has increasingly been applied to physiological data, which often have small sample sizes. Despite studies highlighting the importance of training size for the performance and generalizability of deep neural networks [17, 18], the benefits of automatic feature extraction and complex pattern recognition make these models attractive for challenging fields like biomedical research and human-computer interaction [19, 20, 21]. In fact, many studies have trained deep networks on small-scale datasets and reported high performances compared to shallower models [7, 22, 23]. Our work shows that while this is a feasible approach, as the number of available samples in a dataset decreases, more caution is required during the evaluation of the performance.

Overestimating Model Performance. The performance of an ML model is assessed by its effectiveness on new, unseen, samples. However, tuning hyperparameters and relying solely on validation performance for evaluation can lead to overly optimistic results, while true performance may remain poor [24]. Cawley et al. [8] demonstrated that using cross-validation to tune parameters of a kernel ridge regression classifier leads to a continuous decrease in validation error while test error increases after an initial drop. Similarly, Vabalas et al. [9] and Ambroise et al. [25] show that performance estimates are overly optimistic when cross-validation is used for both model/feature selection and performance evaluation. They recommend using nested cross-validation instead of standard cross-validation to ensure the model is evaluated on an independent test set. Our work extends the previous evaluations to deep learning models and shows that DL greatly amplifies this bias, where even minor model adjustments lead to significant overestimation.

5 Discussion

We have systematically investigated the gap between cross-validation and test performance in neural networks trained on small-scale random data, using early stopping as a regularizer. Our results show that cross-validation performance consistently surpasses chance levels, regardless of variations in training sample size, network depth, and width, data feature size, and optimizers. Notably, with smaller datasets, cross-validation performance can reach up to 95% on random data.

This effect is more pronounced in deeper networks and smaller datasets. While increasing network depth is a common strategy to boost performance, it can also amplify overestimating the performance if evaluation is not carefully managed. Although any information leakage from the test set to the training process skews results, this bias decreases with larger training sample sizes, highlighting the need for caution in low-data scenarios. Interestingly, while increasing network depth significantly increases the validation-test gap, changing network width has little impact on this gap. This can be a result of deep networks being composite functions with several layers of nonlinear transformations applied to the data which can give them greater computational power. Further investigation of this observation using deep linear networks and infinite wide networks is an interesting future work. Another interesting observation is the drop in the cross-validation performance when the network gets very deep (Fig. 4b), probably due to the gradient vanishing (or explosion). Future work could validate this through techniques that mitigate these issues, such as residual connections and gradient clipping.

We used the validation set exclusively for early stopping during neural network training to prevent overfitting, without tuning any other hyperparameters based on it. We demonstrated that even this seemingly minor adjustment can notably skew the estimated model performance based on cross-validation. Even when early stopping is not explicitly applied, researchers often implicitly adjust the number of training epochs based on performance on a non-training sample set. Our study implies that this implicit early stopping can similarly lead to a significant overestimation of performance if an unseen test set is not used.

It is also worth noting that early stopping sometimes includes a patience parameter, allowing optimization to continue for a specified number of epochs after no improvement is observed in the loss. We evaluate the effect of including a patience to early stopping in Section A.6.

Evaluating the influence of other regularization methods, such as weight decay and dropout, on the extent of this overestimation is left for future work. Nonetheless, we showed that for deep learning applications, using any evaluation data during model development could result in highly biased performance estimates. This issue is particularly critical in low-data scenarios, where underestimating its impact can lead to misleading conclusions about a model's effectiveness.

Acknowledgments and Disclosure of Funding

This work was supported by the Knut and Alice Wallenberg Foundation, Swedish Research Council, European Research Council (ERC) D2Smell 101118977 and ERC BIRD 88480 grants.

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 (2012).

[2] A. Dosovitskiy. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[3] A. Vaswani. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017).

[4] T. Teijeiro et al. "Arrhythmia classification from the abductive interpretation of short single-lead ECG records". In: *2017 Computing in cardiology (cinc)*. IEEE. 2017, pp. 1–4.

[5] S. Hong et al. "Practical lessons on 12-lead ECG classification: Meta-analysis of methods from PhysioNet/computing in cardiology challenge 2020". In: *Frontiers in Physiology* 12 (2022), p. 811661.

[6] X. Tang et al. "Motor imagery EEG recognition based on conditional optimization empirical mode decomposition and multi-scale convolutional neural network". In: *Expert Systems with Applications* 149 (2020), p. 113285.

[7] C. Zhang, Y.-K. Kim, and A. Eskandarian. "EEG-inception: an accurate and robust end-to-end neural network for EEG-based motor imagery classification". In: *Journal of Neural Engineering* 18.4 (2021), p. 046014.

[8] G. C. Cawley and N. L. C. Talbot. "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation". In: *Journal of Machine Learning Research* 11.70 (2010), pp. 2079–2107. URL: `http://jmlr.org/papers/v11/cawley10a.html`.

[9] A. Vabalas et al. "Machine learning algorithm validation with a limited sample size". In: *PloS one* 14.11 (2019), e0224365.

[10] A. Blum and M. Hardt. "The ladder: A reliable leaderboard for machine learning competitions". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1006–1014.

[11] R. T. Schirrmeister et al. “Deep learning with convolutional neural networks for EEG decoding and visualization”. In: *Human brain mapping* 38.11 (2017), pp. 5391–5420.

[12] A. H. Ribeiro et al. “Automatic diagnosis of the 12-lead ECG using a deep neural network”. In: *Nature communications* 11.1 (2020), p. 1760.

[13] N. Yeung et al. “Detection of synchronized oscillations in the electroencephalogram: an evaluation of methods”. In: *Psychophysiology* 41.6 (2004), pp. 822–832.

[14] Z. T. Ke, Y. Ma, and X. Lin. “Estimation of the number of spiked eigenvalues in a covariance matrix by bulk eigenvalue matching analysis”. In: *Journal of the American Statistical Association* 118.541 (2023), pp. 374–392.

[15] I. M. Johnstone. “On the distribution of the largest eigenvalue in principal components analysis”. In: *The Annals of statistics* 29.2 (2001), pp. 295–327.

[16] I. M. Johnstone and D. Paul. “PCA in high dimensions: An orientation”. In: *Proceedings of the IEEE* 106.8 (2018), pp. 1277–1292.

[17] C. Sun et al. “Revisiting Unreasonable Effectiveness of Data in Deep Learning Era”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[18] H.-P. Chan et al. “Deep learning in medical image analysis”. In: *Deep learning in medical image analysis: challenges and applications* (2020), pp. 3–21.

[19] O. Faust et al. “Deep learning for healthcare applications based on physiological signals: A review”. In: *Computer methods and programs in biomedicine* 161 (2018), pp. 1–13.

[20] B. Rim et al. “Deep learning in physiological signal data: A survey”. In: *Sensors* 20.4 (2020), p. 969.

[21] D. Xiong et al. “Deep learning for EMG-based human-machine interaction: A review”. In: *IEEE/CAA Journal of Automatica Sinica* 8.3 (2021), pp. 512–533.

[22] A. Vahid et al. “Applying deep learning to single-trial EEG data provides evidence for complementary theories on action control”. In: *Communications biology* 3.1 (2020), p. 112.

[23] S. Tiwari, S. Goel, and A. Bhardwaj. “MIDNN-a classification approach for the EEG based motor imagery tasks using deep neural network”. In: *Applied Intelligence* (2022), pp. 1–20.

[24] T. Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[25] C. Ambroise and G. J. McLachlan. “Selection bias in gene extraction on the basis of microarray gene-expression data”. In: *Proceedings of the national academy of sciences* 99.10 (2002), pp. 6562–6566.

[26] B. Iravani et al. “Non-invasive recording from the human olfactory bulb”. In: *Nature communications* 11.1 (2020), p. 648.

A Appendix

A.1 Searched Parameters

Here, we present different parameters tested in our three experiments with their corresponding values.

Supplementary Table A1: Experiment setup for evaluating the overestimation effect of early stopping on the model performance. We focus on three different components of a standard machine learning pipeline: the *data*, where **N** is the sample size, **Ch** is the number of data channels (i.e., different recording electrodes), **F** is the temporal window size, **D** is the feature size; the *model*, where **A** is the network architecture, **L** is the network depth, **H** is the network width; and the *training* method, where **O** is the optimizer. Note that $[2^x, 2^y]$ means all the values in the set $\{2^i \mid i \in [x, y], i \in \mathbb{Z}^+\}$.

Evaluation	**N**	**Ch**	**F**	**D**	**A**	**L**	**H**	**O**
Real	max. 140 per participant	64	128	-	[ResNet-1D, ConvNet]	-	-	SGD
Synthetic	$50 \times [2^0, 2^5]$	$[2^3, 2^6]$	$[2^6, 2^9]$	-	[ResNet-1D, ConvNet]	-	-	SGD
Gaussian	$50 \times [2^0, 2^6]$	-	-	$[2^2, 2^9]$	MLP	$[2^0, 2^7]$	$[2^2, 2^9]$	[SGD, AdamW]

A.2 Real Data Experiment: Data Description

For the *Real Data* experiment, we employed EEG data from 52 participants, recorded during an olfactory task. EEG Signals were recorded from 64 scalp electrodes while participants were exposed to three neutral odors at two different intensities and a no-odor condition (similar to the process proposed in [26]). Each participant underwent 140 trials, although some were excluded due to noise and artifacts, resulting in less than 140 available samples for most of the participants. Although the original data was collected with 7 different labels (corresponding to different odor types and intensities and the clean air condition), we did not need the labels nor care about the post-stimulus signal which contained task-related information. We selected EEG signals from 600 ms to 100 ms before the stimulus onset to capture baseline activity. The sampling frequency of the data was set to 512 Hz and it was resampled to 256 Hz after a low-pass filtering at 120 Hz. For classification, we randomly assigned half of the signals to class 0 and the other half to class 1, aiming for a baseline classifier performance of 50%. As mentioned in the main text, we used nested cross-validation in this experiment. Each participant's data was split into 10 folds, with one fold as the test set and cross-validation was performed on the remaining 9 folds. The models were trained as explained in Section 2 and tested on the held-out test set. This process was repeated three times with different random seeds, yielding 30 distinct test sets in total.

A.3 Nested Cross-Validation

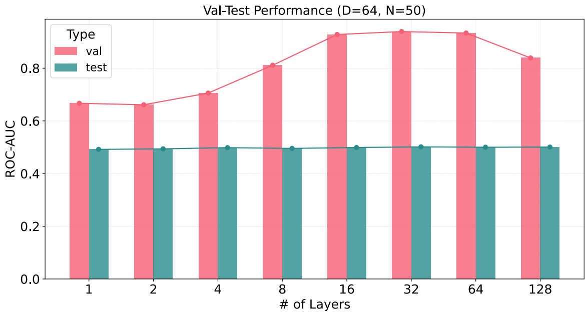
Nested cross-validation was employed to assess both cross-validation and test performance on real EEG data. The method involved two levels of looping: an outer loop and an inner loop.

In the outer loop, the data from each participant was divided into 10 folds. One fold was reserved as the test set, while the remaining 9 folds were passed to the inner loop. In the inner loop, those 9 folds were again split into 10 folds. During each inner loop iteration, one fold was used for validation (to implement early stopping), and the other folds were used for model training. This process continued until every fold had been used once for validation.

The cross-validation performance was calculated by averaging the validation results from all iterations of the inner loop. For test performance, the model was evaluated on the test fold from the outer loop for each different training set, and these results were averaged. The entire process was repeated until each fold in the outer loop had served as the test set once.

A.4 Synthetic Data Experiment: Data Description

For the *Synthetic Data* experiment, we used the model proposed by [13] to synthesize EEG-like activity. The signal was generated by summing 50 sinusoids with randomly varying frequencies and phases, spanning from 0.1 to 125 Hz and phases ranging from 0 to 2π. The amplitude of each



Supplementary Figure A1: Cross-validation and test performance across # of layers for an MLP trained on random Gaussian data with a feature size of 64 and a sample size of 50.

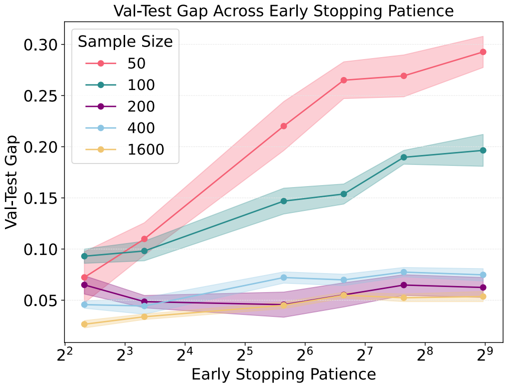
sinusoid was scaled to match the power spectrum of real EEG signals. The data was sampled at 256 Hz, with each epoch consisting of 128 samples, corresponding to 500 ms of signal.

A.5 Severe Performance Overestimation on Small Data

Figure A1 visualizes the cross-validation and the test performance for a sample experiment on random Gaussian vectors. We observe that when the sample size is very small (50 samples here), the cross-validation performance on completely random data can reach up to 95% for deep networks. This is while the true performance (here obtained on 5000 samples from the same distribution) is always 50%.

A.6 The Effect of Patience Parameter

The patience parameter in early stopping allows optimization to continue for a specified number of epochs after no improvement is observed in the validation loss. Higher patience values provide the optimizer with more opportunities to find a better solution. Figure A2 illustrates the validation-test gap across varying patience values, showing that overestimation becomes more pronounced as patience increases.



Supplementary Figure A2: Validation-test gap across varying patience values. Different colors represent different training sample size. The model was an MLP with a hidden size of 16 and depth of 8, trained on random Gaussian vectors with different input feature sizes selected from values provided by Table A1.