Improving Brain Computer Interfaces Using Deep Scale-Invariant Temporal History Applied to Scalp Electroencephalogram Data

Gaurav Anand†
University of Virginia
School of Data Science
Charlottesville, USA
ga7er@virginia.edu

Arshiya Ansari†
University of Virginia
School of Data Science
Charlottesville, USA
aa9yk@virginia.edu

Beverly Dobrenz†
University of Virginia
School of Data Science
Charlottesville, USA
bgd5de@virginia.edu

Yibo Wang†
University of Virginia
School of Data Science
Charlottesville, USA
yw9et@virginia.edu

Brandon G. Jacques
University of Virginia
Department of Psychology
Charlottesville, USA
bgj5hk@virginia.edu

Per B. Sederberg*
University of Virginia
Department of Psychology
Charlottesville, USA
pbs5u@virginia.edu

2021 Systems and Information Engineering Design Symposium (SIEDS) | 978-1-6654-1250-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/SIEDS52267.2021.9483789

***Abstract*—Brain Computer Interface (BCI) applications employ machine learning to decode neural signals through time to generate actions. One issue facing such machine learning algorithms is how much of the past they need to decode the present. DeepSITH (Deep Scale-Invariant Temporal History), is a deep neural network with layers inspired by how the mammalian brain represents recent vs. less-recent experience. A single SITH layer maintains a log-compressed representation of the past that becomes less accurate with older events, unlike other approaches that maintain a perfect copy of events regardless of how far in the past they occurred. By stacking layers of this compressed representation, we hypothesized that DeepSITH would be able to decode patterns of neural activity from farther in the past and combine them efficiently to guide the BCI in the present. We tested our approach with the Kaggle "Grasp and Lift challenge" dataset. This motor movement dataset has 12 subjects, 10 series of 30 grasp and lift trials per subject, with 6 classes of events to decode. We benchmark DeepSITH performances on this dataset against another common machine learning technique for integrating features over extended time scales, long short-term memory (LSTM). DeepSITH reproducibly achieves higher accuracy in predicting motor movement events than LSTM, and also takes significantly fewer epochs and less memory to train, in comparison to LSTM. In summary, DeepSITH can efficiently process more data, with increased prediction accuracy and learning speed. This result shows that DeepSITH is an advantageous model to consider when developing BCI technologies.**

I. Introduction

People who suffer from severe motor disabilities need assistive technologies to interact with their environment [1]. Brain Computer Interfaces (BCI) support this need by allowing users to directly control external devices, e.g. computer, wheelchair, or a neuro-prosthesis, independent of peripheral nerves and muscles [2]. Most BCIs use electroencephalography (EEG) signals as the inputs because of their non-intrusive characteristics [3]. EEG signals are made up of complex spatial and temporal patterns, so successful EEG-based BCI systems are heavily dependent on how much of the past they use to decode the present and on the underlying machine learning model's ability to effectively decode the past in real-time [2].

The goal of this project is to evaluate the utility of temporal event compression on BCI applications by comparing the performance of DeepSITH against LSTM. Using a "Grasp and Lift challenge" dataset consisting of EEG signals for six different hand movements across 12 subjects, we compare the accuracy and training time of DeepSITH (Deep Scale-Invariant Temporal History), a deep neural network with layers inspired by how the mammalian brain represents recent experience, against long short-term memory (LSTM).

LSTM is a favored machine learning approach because of its ability to process entire sequences of data as opposed to relying on short buffers of data points. However, LSTM models are limited by the amount of data they can process because they must learn to maintain an exact representation of relevant features from the past to guide decisions in the present. Unlike LSTM, DeepSITH maintains a log-compressed representation of the past that, in cases where the time scale of relevant information is unknown, provides an optimal fuzzy integration of the past [4]. Although the SITH representation decreases in temporal precision the farther an event occurred in the past, it also compresses the signals such that more data can be used to inform the model. By comparing DeepSITH with LSTM, we demonstrate the tradeoff of memory requirements vs. event description accuracy and inform the utility of including DeepSITH in the development of BCI technologies.

† These authors contributed equally to this work.
* Corresponding author.

978-1-6654-1250-6/21/$31.00 ©2021 IEEE

II. Background

A. Summary of BCI + EEG data

Traditional BCI systems consist of several components: brain signal acquisition, signal preprocessing, feature extraction and classification through machine learning, and the translated physical action [5]. EEG signals are the most common brain signals in BCI applications because they are relatively non-invasive. The method of collecting EEG signals involves attaching conductive electrodes to a subject's scalp to measure the voltage change resulting from ionic activity within neural populations. While non-invasive and extremely common, the EEG signal quality is rather poor and intensive signal preprocessing such as signal filtration or signal reduction is often required for physical actions to be properly translated [6].

B. Current BCI techniques

In previous BCI applications, signal-to-action classification has employed simple machine learning techniques such as linear classifiers, Bayesian classifiers, artificial neural networks, and ensemble methods involving all three [7]. As mentioned before, the noise associated with EEG signals makes them difficult to decode with just feature extraction or data preprocessing methods due to the tremendous time complexity and the risk of information loss. Common techniques for multi-class motor action classification with EEG data generally yield accuracy below 80% due to the dynamic and noisy signal. Therefore, researchers have shifted to investigate deep learning techniques, which have proved to be more powerful than the conventional classifiers mentioned [6].

C. Deep learning methods

Deep learning is a special area of machine learning in which features and the model parameters are learned directly from the data and are less reliant on time-consuming preprocessing and feature extraction steps necessary for complex data [7]. The field of deep learning was inspired by the structure and function of the brain; artificial neurons are the simple building blocks that communicate with one another to create complex neural networks, which are capable of processing large amounts of data and detecting patterns for use in decision making. Deep neural networks have shown increased success in addressing the challenges of low classification accuracy because they can capture both high-level features and underlying dependencies in the EEG signals through their non-linear and deep structures [6].

D. Deep learning and long-range temporal dependencies

The main challenge with deep neural networks is the perfect storage of all historical data for accurate model predictions. Perfect storage is currently impossible as it is extremely expensive to store the data and equally costly to train a model. Usually the cost of memory storage must be balanced with model prediction accuracy, therefore, less historical data stored entails a less accurate model.

One popular current approach to represent historic data is maintaining a fixed buffer size of past data, which uses a first-in first-out mechanism to store relevant historical data. A fixed buffer constrains the available temporal interval over which detailed event information can be stored [8]. In addition, the fixed buffer also generates deep networks that ignore the mutual information between long-range time points, causing the overall accuracy of the model to decrease [9].

Neural networks have improved in terms of enabling an accurate storage of information over time, especially with the emergence of recurrent neural networks (RNNs) and long-short term memory networks (LSTMs). Unfortunately, these network structures still face the issue of exploding/vanishing gradients, which is when a time series is fed to the network, and then the gradient of the loss function is calculated at the end of the sequence [9]. After each sequence, the gradient will either increase or decrease via backpropagation. An exploding gradient occurs when the gradient increases exponentially, and a vanishing gradient occurs when the gradient decreases exponentially. This is a current issue with RNNs because a vanishing gradient acts similarly to a small buffer, in that information from the distant past does not have a significant effect on network weights regarding a current prediction. This means that information from too far into the past will likely be ignored by the network when making a prediction [10]. LSTMs attempt to address the exploding/vanishing gradient problem by encoding long-range dependencies and relations in the LSTM cell state vectors. However, LSTMs are not successful when the time scale is too long-range and the LSTM model parameters must be continuously adjusted to learn the relevant time scale [9].

E. Scale invariant techniques and SITH

To address the problem, researchers have looked at the way the human brain maintains a logarithmically-compressed representation of past experience for inspiration. Prior work in several modalities, from delayed match to sample reinforcement learning, has demonstrated that time is represented in the brain in a scale-invariant fashion [11] [12]. Scale invariant features, such as events recorded in time, do not change if the time scale of the event is dilated or compressed by a scalar factor. That is, regardless of how much the scale of time is zoomed in or out, the relationship between events represented at any particular scale will not change [13]. In the hippocampus, individual time cells show spiking activity at specific temporal intervals. They are sequentially-activated time cells and exhibit behavior that suggest time-scale invariance. When graphed, the firing fields of time cells that fire later in a delay period are wider than the firing field of time cells that fire earlier in the delay period, meaning that the further in the past, the lower the temporal specificity becomes. Scale invariance of episodic representation in the brain provides the basis for an individual to use the same set of mechanisms to integrate information and make decisions over different time scales.

Scale-Invariant Temporal History (SITH) is built on the concept of time being represented in a scale-invariant fashion and may be a better alternative to both fixed length buffer and LSTM approaches. When given a time series as input,

SITH encodes an approximation of the input via a family of logarithmically-spaced impulse response functions representing temporal receptive fields centered on times τ^* with widths that increase as a function of time into the past (see bottom of Fig. 1). SITH was first tested within an artificial intelligence framework on the video game Flappy Bird, where the model was trained via deep reinforcement learning to play variants of the game with and without long temporal information gaps. It proved to be surprisingly successful in spanning long time periods with no meaningful input, whereas fixed buffers with the identical memory use failed to learn to span these temporal gaps [8]. DeepSITH extends this original approach to a deep neural network architecture made up of many SITH layers, greatly improving the time scales over which it can integrate information while still overcoming the issue of exploding/vanishing gradients by providing a complete state of the history to the model at every time step [9].

III. Methodology

A. Dataset

To identify relationships and patterns that occur between EEG signals collected from the brain and hand movements collected from participants, the WAY consortium organized the 2015 Grasp-and-Lift EEG Detection Challenge. The dataset from the competition contains scalp EEG data sampled at a frequency of 500 Hz from over 12 participants that conducted over 300 tasks. The outcome of the challenge was being able to detect 6 events that represented different hand movements that were made by the participants. [14]

We divided every series into 4-second (i.e., 2000-sample) segments with a step size of 2 seconds (1000 samples). For the purposes of building our model, the dataset was broken down into a training and validation split for each subject, with the first 80% of the subject batches being used for training and the remaining 20% being used for testing and validating model. We note that this approach to a training and test split may result in the generation of test samples that could be slightly dependent on data used in the training set, possibly inflating AUC values relative to an independent test set for the original competition. Thus, we focus below on the comparison of models trained and tested under identical conditions.

B. Preprocessing

One challenging aspect of working with EEG data is the amount of noise or artifacts present in the data. These artifacts are simply undesirable signals that originate from the environment and contaminate the quality of the EEG signal. Furthermore, the intensity of EEG signals captured by the sensors varies across the electrodes in the scalp EEG setup. This means that preprocessing steps need to be taken before event detection modelling can take place. The two main preprocessing tools that we employed were filtering and standardization. Filtering is one of the most widely used preprocessing techniques used for activity detection. [15] However, one of the conditions of the challenge associated with the dataset was ensuring that no data from the future was used when training the model. In order to eliminate the undesirable signals, we applied a minimum phase low pass FIR filter at 30 Hz from the MNEtools package to isolate grasping frequencies. [16] The use of a causal filter ensured that data from the future was not used to help our model learn in order to mimic real-time BCI preprocessing strategies. Furthermore, a simple z-score standardization was also performed to compare the signals from different channels on the same scale. The preprocessing approaches were implemented using a 2 second or 1000 sample wide sliding window with a step size of 1 sample.

C. DeepSITH Architecture

DeepSITH architectural layers extract information at different temporal scales. Each DeepSITH layer tracks the histories of an arbitrary number of input features. Each SITH layer outputs the activation levels of $N_\tau * N_{features}$ log-spaced τ^* values representing the peak of the temporal receptive fields. With these activation levels, the network has access to the conjunctive representation of "what happened, when" in a reduced form. A set of τ^*s exist for each feature, and is passed through a linear layer that allows the network to make associations between features and time. The output of this hidden layer, with a ReLU activation function, is then passed as the input into the next DeepSITH layer. After the last DeepSITH layer, a linear layer then transforms the DeepSITH output into the size appropriate for the task at hand.

Each DeepSITH layer has six hyperparameters that need to be tuned for each specific task. These are τ_{min}, τ_{max}, k, dt, N_τ, and N_{hidden}. N_{hidden} dictates the size of the output of the linear layer after the SITH layer. The other five hyperparameters dictate the number and size of the τ^* temporal receptive fields. Normally, we choose $tau_{min} = dt = 1$, and we have τ_{max} increase logarithmically. k is chosen via a special minimization function dictated in [9]. N_τ is the same on each layer, but chosen to be a small number.

D. DeepSITH Network and Tuning

Consistent with the work of Jacques et al. [9], we used a PyTorch [17] implementation of the SITH layer that approximates the temporal history of the input function $f(t)$ through the use of a discrete approximation to the inverse Laplace transform. By finding the approximate values of $f(t)$ at any scale, we are able to utilize scale-invariant reconstructions of the temporal relationships within the input feature in our deep learning algorithm.

DeepSITH network layers are governed by a few key hyperparameters - τ_{min}, τ_{max}, k, dt, N_τ and g - which we needed to tune in order to optimize our results. The τ_{min} and τ_{max} variables identify the center of the receptive fields for the first and last τ^* values, respectively, within the ensemble of reconstructed timescales. Together, the two variables define the range of the receptive fields. The value of k identifies the temporal specificity or the sharpness of the receptive fields, with larger k giving rise to more narrow impulse response functions. dt represents the time delta of the model. N_τ sets

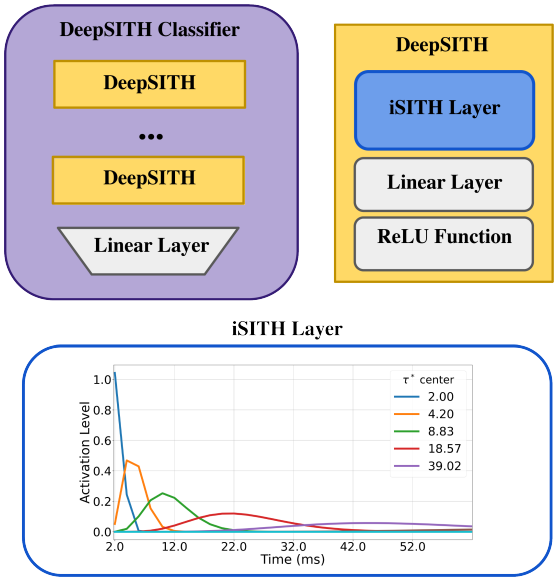

Fig. 1. *A visualization of the DeepSITH classifier network.* The DeepSITH network used here is made up of several DeepSITH layers which consist of a SITH layer followed by a linear layer and an activation layer. Adapted from [8]. The bottom panel shows impulse responses of 5 SITH filters for the last layer of our DeepSITH model.

the size of the ensemble and finally g sets the scale of the model. Finally, we can also define the number of hidden layers which can also be thought of as approximately the number of associations that are extracted from the temporal history of the decomposed SITH layer input features.

For the purposes of event detection, we tune our DeepSITH network parameters using the methodology suggested by Jacques et. al [9]. The τ_{max} was set to logarithmically increase from layer to layer while the τ_{min} was kept constant. The network was also found to perform optimally at 3 layers in depth with each layer having a hidden layer of size of 20. The value of k was calculated at each layer by finding the value that minimized the ratio of the standard deviations to the sum of all scale-invariant filters and the sum of alternating scale-invariant filters. By calculating the k value in this manner, we were able to make sure that the amount of information stored from the past decays with time. This optimization of the temporal specificity of the scale-invariant filters allows us to represent the impulse response at each reconstructed timescale, τ^* such that each response shares information while preventing unnecessary overlap between them. The number of receptive fields in the model, N_τ, was kept low to sufficiently decrease model complexity. Furthermore, a 10% drop-out layer was added to increase model generalizability. The total number of tunable parameters was 14,000.

E. LSTM Network

For evaluation against the DeepSITH Network, we used a 3-layer deep LSTM network along with a 10% dropout layer for improved model generalizability. Each LSTM layer was chosen to have 25 hidden nodes in an effort to obtain a similar number of tunable parameters to the DeepSITH network. The LSTM network contained 17,000 tunable parameters.

TABLE I
DEEPSITH PARAMETERS

	Layer 1	Layer 2	Layer3
τ_{min}	1	1	1
τ_{max}	50	200	800
k	23	12	7
dt	1	1	1
N_τ	10	10	10
g	0	0	0
hidden size	20	20	20

F. Model Evaluation

We evaluated DeepSITH and LSTM performance on four different metrics, namely mean column-wise area under receiver operating characteristics curve (AUC), precision, recall and F1 scores (Table II). Mean column-wise area under receiver operating characteristics curve (AUC) is the official evaluation metric for the EEG Grasp-and-Lift Kaggle competition. Receiver operating characteristics (ROC) shows the changes of the true positive rate with respect to the false positive rate, and the AUC takes into consideration all thresholds, showing an objective classification accuracy for the model regardless of the choice of threshold. Precision, recall and F1 scores can provide an intuitive grasp on the performance of these models on real-world BCI applications. Specifically, precision measures the true positives amongst all of the positive classifications made by the classifier. This provides insight into our classifier's ability to accurately distinguish the positive class. Recall measures our classifier's ability to discriminate between both the positive and negative classes against the entire set of positive classes. F1 score is then the harmonic mean of precision and recall. We picked a threshold of 0.3 to binarize the probability outcomes derived from the models, which gives relatively high F1 score for both models.

article array booktabs

IV. RESULTS

In order to test the training speed of DeepSITH, we compared a DeepSITH model with a similarly structured LSTM model. The LSTM model we used had the same number of layers and training hyperparameters. The LSTM model also had slightly more total number of tunable parameters than DeepSITH (DeepSITH: 14,000, LSTM: 17,000), since we want to make sure both models have enough capacity to learn the complex tasks. DeepSITH could achieve significantly higher validation accuracy than LSTM after only a few epochs of training (Fig. 2.). In fact, DeepSITH on average only takes 2 to 3 epochs to achieve 0.8 validation AUC as indicated by the purple dashed line shown in the figure, whereas it takes significantly longer for LSTM to reach the 0.8 level.

DeepSITH also achieved significantly higher AUC for every subject than LSTM. We then evaluated DeepSITH performance on precision, recall and F1 scores, which require

TABLE II
SUBJECT-LEVEL PREDICTION FOR DEEPSITH AND SIMILARLY STRUCTURED LSTM

Subject	AUC		Precision		Recall		F1	
	DeepSITH	LSTM	**DeepSITH**	LSTM	**DeepSITH**	LSTM	**DeepSITH**	LSTM
1	**0.997**	0.967	**0.813**	0.597	**0.905**	0.722	**0.856**	0.647
2	**0.980**	0.912	**0.697**	0.424	**0.743**	0.444	**0.715**	0.421
3	**0.990**	0.913	**0.768**	0.498	**0.834**	0.541	**0.798**	0.517
4	**0.992**	0.960	**0.788**	0.599	**0.853**	0.645	**0.818**	0.619
5	**0.987**	0.913	**0.749**	0.444	**0.802**	0.374	**0.774**	0.378
6	**0.991**	0.951	**0.729**	0.530	**0.815**	0.550	**0.769**	0.526
7	**0.989**	0.955	**0.697**	0.505	**0.795**	0.568	**0.737**	0.525
8	**0.983**	0.927	**0.731**	0.532	**0.794**	0.568	**0.760**	0.543
9	**0.992**	0.962	**0.767**	0.537	**0.833**	0.644	**0.798**	0.582
10	**0.991**	0.944	**0.749**	0.486	**0.815**	0.583	**0.778**	0.512
11	**0.981**	0.938	**0.744**	0.557	**0.732**	0.601	**0.737**	0.571
12	**0.987**	0.936	**0.674**	0.510	**0.733**	0.539	**0.702**	0.517

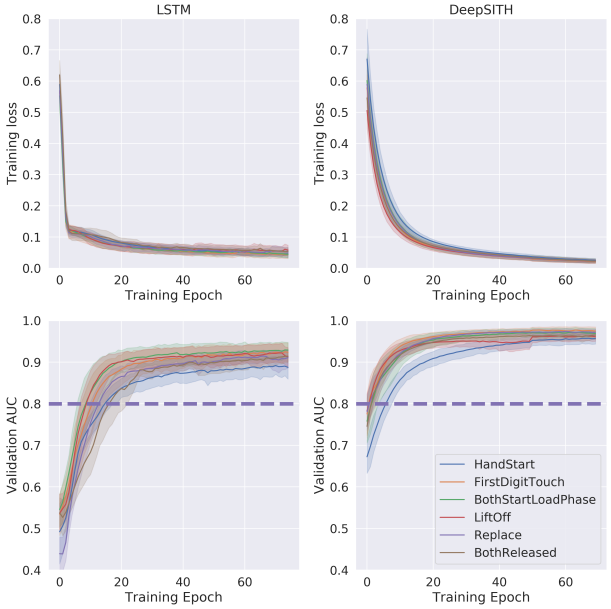

Fig. 2. *Loss and validation AUC curve for subject-level training.* The mean and 95% confidence interval for each event across all subjects are shown. The purple dashed lines indicate validation AUC of 0.8. Each epoch consisted of one pass over the entire training set for a particular subject and event.

a hard threshold on the final probability outputs. Due to the imbalanced nature of this dataset (approximately 1 to 50), it remains challenging to accurately classify every time point without sacrificing either precision or recall. LSTM can achieve great prediction AUC that is similar to what was previously reported, but precision, recall and F1 remain low for most of the subjects (TABLE II). In contrast, we show that precision, recall and F1 are all in the 70% to 80% range with a single threshold of 0.3 for DeepSITH. For further comparisons with other public methods on this dataset, we will submit our predictions to Kaggle to get a final AUC on the public test dataset.

V. DISCUSSION

BCI applications require powerful technological advances to assist individuals with disabilities through multiple aspects of their lives. The success of these technologies is heavily dependent on a robust, underlying algorithm that can accurately generate predictions while efficiently incorporating past information to decode intentions in real-time. The analysis detailed in this paper focuses on evaluating training time and accuracy values across two different BCI learning approaches, DeepSITH and LSTM. We conclude that adding DeepSITH layers improves training time and accuracy values over LSTM models, but also want to note that there are other metrics that can further inform DeepSITH's impacts on a machine learning model.

For our analysis we primarily focused on accuracy, measured by AUC. AUC is indifferent to class imbalances so it was not impacted by a skewed dataset. Different BCI applications can be informed by data with varying skewness levels, so it is important to further understand the effects of a skewed dataset on DeepSITH performance. To provide this insight we recommend that future research evaluate changes to precision and recall. In a physical sense, low precision means that a model labeled negative samples positive (the BCI technology grasps an object but the user did not intend to grasp it). Low recall means that a classifier is not able to identify all of the positive samples, resulting in more false negatives (a user wants to grasp an object but the BCI technology does not respond). Understanding how precision and recall values vary when DeepSITH layers are introduced will inform a more holistic approach to improving BCI technologies. Furthermore, it is important to duplicate the analysis of DeepSITH's effects on training time, accuracy, precision, and recall against other EEG datasets. From motor movement to motor imagery, the identification of a machine learning approach that can successfully and efficiently generate accurate predictions across a diverse range of applications can inform and improve the BCI field as a whole.

ACKNOWLEDGMENT

We thank Prof. Timothy Clark, and Dr. Sadnan Al Manir from the University of Virginia for their guidance and support throughout this project.

REFERENCES

[1] L. Qin, L. Ding, and B. He. Motor imagery classification by means of source analysis for brain–computer interface applications. *Journal of Neural Engineering*, 1(3):135–141, 2004.

[2] S.-M. Zhou, J. Q. Gan, and F. Sepulveda. Classifying mental tasks based on features of higher-order statistics from eeg signals in brain–computer interface. *Information Sciences*, 178(6):1629–1640, 2008.

[3] C. Tan, F. Sun, and W. Zhang. Deep transfer learning for eeg-based brain computer interface. In *2018 IEEE International Conference on Acoustics*, pages 916–920, Calgary, AB, Canada, 2018. Speech and Signal Processing (ICASSP).

[4] Karthik H. Shankar and Marc W. Howard. Optimally fuzzy temporal memory. *Journal of Machine Learning Research*, 14(83):3785–3812, 2013.

[5] X. Zhang and D. Wu. On the vulnerability of cnn classifiers in eeg-based bcis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(5):814–825, 2019.

[6] X. Zhang, L. Yao, X. Wang, J. Monaghan, and D. Mcalpine. A survey on deep learning based brain computer interface: Recent advances and new frontiers. *arXiv.org*, 10, May 2019.

[7] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. A Yger. A review of classification algorithms for eeg-based brain-computer interfaces: a 10 year update. *J Neural Eng.*, 28, June 2018.

[8] T. A. Spears, B. G. Jacques, M. W. Howard, and P. B. Sederberg. Scale-invariant temporal history (sith): optimal slicing of the past in an uncertain world. *arXiv.org*, 17, December 2017.

[9] Brandon Jacques, Zoran Tiganj, Marc W. Howard, and Per B. Sederberg. Deepsith: Efficient learning via decomposition of what and when across time scales, 2021.

[10] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[11] Howard Eichenbaum. On the integration of space, time, and memory. *Neuron*, 95(5):1007–1018, 2017.

[12] K. H. Shankar and M. W. Howard. A scale-invariant internal representation of time. *Neural Comput*, 24(1):134–193, Jan 2012.

[13] Marc Howard and Howard Eichenbaum. The hippocampus, time, and memory across scales. *Journal of experimental psychology. General*, 142, 08 2013.

[14] M. D. Luciw, E. Jarocka, and B. B. Edin. Multi-channel eeg recordings during 3,936 grasp and lift trials with varying weight and friction. *Scientific Data*, 1, 2014.

[15] Xiao Jiang et al. Removal of artifacts from eeg signals: A review. *Sensors*, 19:5, 2019.

[16] A. Gramfort, M. Luessi, E. Larson, D. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. Hämäläinen. Mne software for processing meg and eeg data. *NeuroImage*, 86:446–460, February 2014.

[17] Adam Paszke et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32. Curran Associates, Inc.