EEG Channel Interpolation Using Deep Encoder-decoder Networks

<anonymous author>[1], < anonymous author >[1], <anonymous author>[1], and <anonymous author>[1]

Abstract— **Electrode "pop" artifacts originate from the spontaneous loss of connectivity between a surface and an electrode. Electroencephalography (EEG) uses a dense array of electrodes, hence *popped* segments are among the most pervasive type of artifact seen during the collection of EEG data. In many cases, the continuity of EEG data is critical for downstream applications (e.g. brain machine interface) and requires that popped segments be accurately interpolated. In this paper we frame the interpolation problem as a self-learning task using a deep encoder-decoder network. We compare our approach against contemporary interpolation methods on a publicly available EEG data set. Our approach exhibited a $\sim 10\%$ improvement over the contemporary approaches when tested on subjects and tasks not used during model training. We demonstrate how our model's performance can be enhanced further on novel subjects and tasks using transfer learning. All code and data associated with this study is open-source to enable ease of extension and practical use. To our knowledge, this work is the first solution to the EEG interpolation problem that uses deep learning.**

I. INTRODUCTION

Electroencephalography (EEG) devices have become increasingly popular in recent years and are used in a wide range of applications. Naturally, the medical applications of EEG are centered on neurological diagnosis, but EEG has proven useful for other problems in healthcare domain [1], [2]. Moreover, the use of EEG devices extends far beyond the medical domain; novel applications of EEG may be found in wide a variety of fields including advertising [3], education[4], entertainment [5], and security [6].

A foundational challenge of EEG data is the low signal to noise ratio. Different sources contribute to this noisiness but, in general, they can be categorized as either movement artifacts or electrode artifacts. The most common, and particularly persistent, electrode artifact is the electrode *"pop"* [7], [8]. These artifacts result from abrupt changes in impedance, usually due to a loose electrode or bad conductivity. Furthermore, these artifacts are difficult to avoid because, even if the greatest care is taken when applying electrodes, the most minor subject movement or change in perspiration can cause the electrode to "pop".

A common solution to EEG *"pops"* is to interpolate the missing segments using recordings from nearby electrodes [9]. In practice, this interpolation is most commonly performed using *eeglab*, which contains a tool for spherical interpolation [10], [11]. Within the last few years, alternative interpolation methods reporting improved performance have been proposed: Petrichella *et al.* proposed a euclidean inverse distance method [12] while Courellis *et al.* demonstrated an interpolation approach that (while also being based on an inverse distance calculation) used geodesic lengths and electrode localization to extract more exact channel locations, thereby performing more accurate interpolation. [13].

One shortcoming of these previous solutions is their dependence on knowledge of the precise locations of the electrodes (i.e. electrode localization / registration), which are not collected in most practical settings. Furthermore, the existing methodologies assume that the incidence and specific characteristics of the *"pops"* are similar across both subjects and tasks (i.e. "one-size-fits-all"). Furthermore, as far as the authors are aware, none of the studies surveyed for the purposes of this work provided publicly available software repositories to enable practical use, reproduction of their methodologies, or ease of extension.

To address the aforementioned challenges, we propose a novel electrode interpolation framework using representation learning. Our method autonomously identifies the spatio-temporal properties of EEG data measured at a set of electrodes, that predict the values of a given neighbor to those electrodes. Our model, which has been made publicly available [1], can be used "out of the box" to more effectively interpolate EEG for any missing channel, at any time.

One important advantage of our model over existing approaches is its amenability to transfer learning: the ability to easily fine tune it using clean data from a novel subject or EEG experiment. This property of our model allows for interpolation that is tailor-made to the specific task and subject at hand, enabling the model to learn even idiosyncratic relations in new data.

To determine the usefulness of our method we evaluate the model on unseen tasks and subjects with and without further tuning. To summarize, our main contributions are:

- We propose and implement a new framework for EEG channel interpolation using encoder-decoder deep representation learning.
- We compare our method against contemporary algorithms for channel interpolation.
- We make our code publicly available for the benefit of the community, and demonstrate how it may be tuned to novel subjects and tasks using transfer learning.

II. RELATED WORK

A. Current Interpolation Solutions

The Spherical interpolation method (as implemented by *eeglab*) was first proposed by Perrin *et al.* in their 1989 work [10] and remains the most widely-used interpolation

[1] <Anonymous Department>, <Anonymous University>, <anonymous address> <anonymous email>

[1] https://github.com/<url to be released following publication>

solution [10], [11], [14], [12]. More recent work describes improved interpolation methods by, for instance, using ellipsoid geodesic lengths [13] to better estimate the distances between electrodes. A weighted signal reconstruction scheme that favors electrodes closer to the location of the missing channel is then used to achieve higher accuracy. This however requires that the electrode positions be digitally registered so that an ellipsis can be fitted to the shape of the subject's head. That ellipsis is then used when calculating the distances between electrodes. Unfortunately, most data tends not to provide the specific channel locations.

B. Artifact Detection Using Neural Networks

EEG is a signal with spatial structure that unfolds in time. Convolutional neural networks (CNNs) are an obvious candidate for the EEG interpolation problem because they naturally capture hierarchical spatio-temporal relationships.

A few recent papers have leveraged CNNs for EEG classification [15], [16]. Previous works have also used CNNs to annotate EEG wave-forms for the presence of artifacts: Nejedly *et. al.* framed artifact detection as a classification problem and used CNNs to robustly annotate eye blink segments [17]. However, while the use of deep learning approaches for artifact *detection* has shown promising results, there is relatively less work on the use of deep learning to *reconstruct* artifact-ridden data segments.

To our knowledge, our work represents the first use of deep learning for signal reconstruction of EEG data. That is, our work is distinguished from prior artifact removal efforts by using neural networks as a component of a *generative*, rather than a *discriminate*, model. This distinction is especially meaningful in the context of EEG data, where merely annotating and removing noisy segments is not feasible for the many applications that require signal continuity. Generative frameworks can be used to fill in the data after it was removed, allowing researchers and practitioners alike to maintain the data sample even if one of the 20 (or more) channels had bad impedance.

III. Data

A. Data Collection

The data in this study was collected from the recently published *EEG During Mental Arithmetic Data Set* [18]. The data consists of 24 subjects performing two tasks: a *resting state task*, and a *mental arithmetic task*. EEG data was collected as subjects performed the tasks using the 10-20 international system with the linked ears serving as a reference electrode (see Figure 2, left). The sampling rate was $500Hz$. Each *resting state task* lasted 180 seconds while each *mental arithmetic task* lasted for 60 seconds; hence, the total number of samples was $90,000$ and $30,000$ for each *resting state* and *mental arithmetic* task respectively.

We segmented the data into $16ms$ (8 samples) epochs. Hence, following segmentation, each subject had $11,250$ and $3,750$ epochs for the *resting state task* and *mental arithmetic task* respectively.

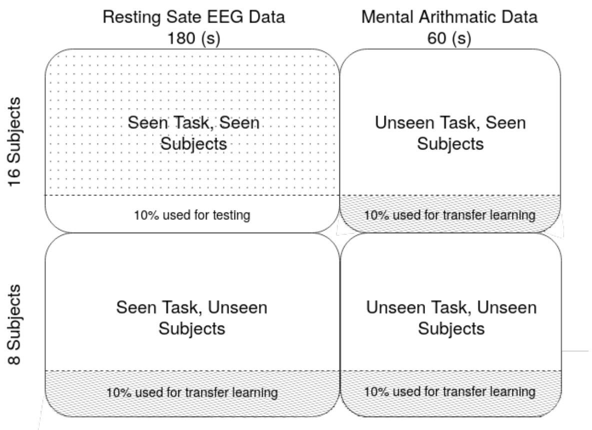
Resting Sate EEG Data
180 (s)
Mental Arithmatic Data
60 (s)
16 Subjects
8 Subjects
Seen Task, Seen Subjects
10% used for testing
Unseen Task, Seen Subjects
10% used for transfer learning
Seen Task, Unseen Subjects
10% used for transfer learning
Unseen Task, Unseen Subjects
10% used for transfer learning

Fig. 1. The four-way split of the data into partitions. The dotted area (top left) was used to train our main algorithm. The areas with the diagonal shading were used for transfer learning. The areas with white background were used to test our algorithm. Each shaded area was used separately to tune the model for testing on the remaining 90% of the data in its own partition.

B. Data Partitioning

In Figure 1, we illustrate how the data was partitioned for the training and evaluation of our method. Training data from the *resting state task* was used for model development while data from the *mental arithmetic task* was held out for intra-task evaluation. We further partitioned data from the both tasks into training subjects (67% of subjects) and evaluation subjects (33% of subjects). This resulted in the following four partitions of the data, which we will refer to later when discussing our results:"*Seen Task, Seen Subjects*" (n=16),"*Seen Task, Unseen Subjects*" (n=8), "*Unseen Task, Seen Subjects*" (n=16), and "*Unseen Task, Unseen Subjects*" (n=8).

The "unseen" data sets contain some deviation from the data the model was trained on, and are thus a good test for the generalizability of the method. Because we are utilizing transfer learning 10% of the data for all unseen sets was held-out for tuning, to test the impact of this additional context on the model's performance.

In general, the fundamental difference between the two tasks is crucial to our evaluation. Researchers will often not have enough data to train neural networks "from scratch", not to mention training a neural network often requires exploration of a hyper-parameter space that may consume significant temporal (and financial) resources. This is especially true for deep learning frameworks that have strong tendencies to over fit and produce remarkable results for a specific data set while failing to generalize to other contexts. With this in mind, it is important that our models achieve good results on both unseen tasks and subjects to enable their continued development and utility within the greater research community. We therefore structured our data to assess its ability to generalize across tasks and subjects.

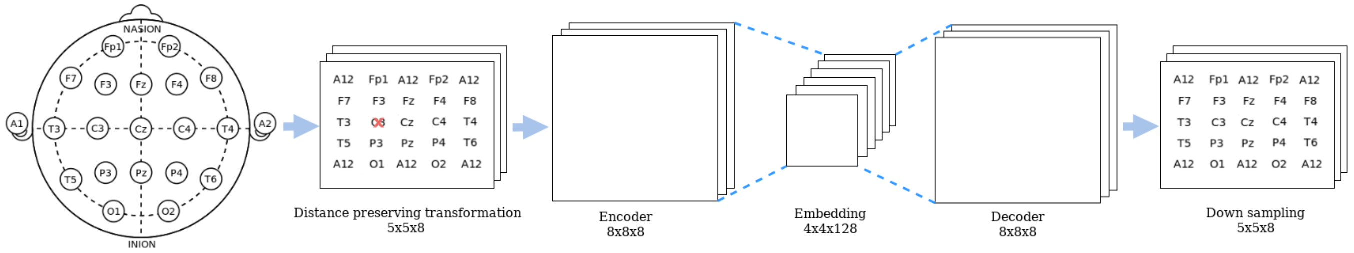

Fig. 2. A diagram of our deep learning framework. The EEG data is first segmented into $16ms$ epochs with each segment mapped to a $5x5$ matrix that roughly reflects the spatial locations of the EEG electrodes (e.g. F7 is located in position 1,1 of the matrix). The electrodes at the sagittal and median planes were duplicated and the tensor was padded with the linked ear channel data to create a 8×8 matrix. We segmented the $500Hz$ data into $16ms$ windows, yielding a final $8 \times 8 \times 8$ input data tensor. This data serves as the input to an encoder-decoder model.

IV. Methods

A. *Pre-processing*

To begin, all EEG data were Z-scored at the subject-level (i.e. converted to a zero mean, and unit variance representations). Deep networks require a large volume of training data; hence, we supplemented our training data by transforming each subject's EEG data into 10 distinct *pseudosubjects*. Each pseudosubject's data was an elementwise addition of the original subject's EEG data and random draws from a Gaussian distribution with ($\mu = 0$, $\sigma = 0.05$). The pseudosubject's EEG data was Z-scored following the introduction of the noise. Next, a simple transformation was applied to project each sample of EEG data from a spherical channel representation onto a quantized two dimensional surface represented by a 5×5 matrix (Figure 2, panel 2).

Finally, EEG data was epoched into $16ms$ segments, with no overlap across segments. This resulted in a 5 channels $\times 5$ channels $\times 8$ samples tensor. The electrodes at the sagittal and median planes (the central electrodes) were then duplicated and the tensor was padded with the linked ear channel data to create a $8 \times 8 \times 8$ tensor. This manipulation of input size is common place in deep learning and is mainly the result of networks being optimized to work with input sizes that are powers of two [19]. This $8 \times 8 \times 8$ tensor formed formed a single sample of input data, from the perspective of the network when training.

To create training data, we iterative occluded each of the 19 non reference electrodes (All electrodes except $A1$ or $A2$, see Figure 2). Thus each $8 \times 8 \times 8$ tensor became the prediction target for 19 input tensors, each with one distinct occluded channel.

B. *Proposed Approach*

1) An Encoder-Decoder model for EEG Interpolation: Inspired by research on image inpainting we deployed an encoder-decoder model for EEG interpolation. Image inpainting is a classical problem in computer vision: given a corrupted image the aim is to complete or "fill in" missing pixels. This is a similar problem to electrode interpolation.

Encoder-decoder models are the combinations of two networks that are trained simultaneously: the encoder first learns a lower dimensional embedding of the data, and the decoder attempts to recover the original data from the embedding. Encoder-decoder networks are a popular tool in image inpainting [20], [21], [22], so much so that this technique is now leveraged for image compression as selective removal of pixels might greatly enhance compression ratios [23].

We determined the optimal topological configuration of our encoder-decoder network via a random search of the network hyper-parameter space [24]. The tested topologies varied in the number of convolution layers, the existence of max-pooling, dropout, and batch normalization layers after each convolution layer, and whether the decoder was based on transposed convolution layers or simple up-sampling with convolution. More specifically, we trained 300 distinct architectural configurations of the encoder-decoder networks, and retained the configuration that best generalized within a held out subset of the training data itself. The best network was then used after training for our transfer learning evaluation. The code to run this search in the topological space, as well as the trained winning algorithm before and after the tuning using transfer learning is available online. The optimal topology is shown in Figure 3, and discussed in Subsection V-A.

2) Subject+Task Enhancement via Transfer Learning: Transfer learning experiments were carried out by taking the model trained on the original data set and tuning it on a small subset (10%) of the testing data. Realistically, it is highly likely that a small sample of clean data will be available for a researcher to use when tuning our model.

To assess performance enhancements associated with transfer learning, we held out 10% of each data partition (See Figure 1) and tuned our network for 100 epochs. These numbers were intentionally small as to showcase how even minimal training that can be easily completed on non specialized hardware and using very little data can lead to significant improvements. By tuning the model for specific subjects and tasks, we assessed the flexibility and practical extensibility of our proposed approach.

C. *Methodological Baselines*

For our baseline we implemented the three methods described in the Related Work section [10], [12], [13].

1) The Euclidean Baseline: Both [12], [13] suggest methods that employ an inverse distance metric where the interpolated channel $\hat{s}_i$ is calculated using the following equation:

$$\hat{s}_i = \frac{\sum_{j \neq i} w_{ij} s_j}{\sum_{j \neq i} w_{ij}}, w_{ij} = \frac{1}{d_{ij}^p}$$

where p is the power parameter. The variable d_{ij} represents the distance between electrode i and electrode j. The original channel j is represented by s_j. The power parameter is an integer (usually between 2 and 5) that is set using a small amount of the data; while it is usually set to be the same value across a given data-set where interpolation is happening, we optimized the power parameter separately for each baseline and data-set to maximize the performance of the baselines.

The calculation of the distance, d_{ij}, is the main difference between the two baselines. The fist euclidean baseline (*EUD*) uses a simple euclidean distance formula. This distance calculation is done using the generic electrode positions in space that are always available for every cap.

2) The Geodesic Baseline: The Geodesic Length baseline (*EGL*) is also based on the inverse distance equation. However, instead of using euclidean distances for d_{ij} the geodesic length is calculated. The geodesic distance is calculated using the *Vincenty* algorithm which was originally used in geodesy to calculate the distance between points on the surface of a spheroid. The method is iterative and not theoretically guaranteed to converge. Previous work have demonstrated that interpolation calculated using this method outperforms simple euclidean interpolation [13]. However, as previously discussed, this was tested by the original baseline works when specific electrode locations were available [13]. It should therefore be expected that results obtained for the *EGL* may be lower than those reported in previous studies.

3) The Spherical Splines Baseline: Finally, we also followed the *eeglab* MATLAB implementation of spherical splines method (*SS*) [10]. According to this implementation at each point in time the value of of the interpolated channel $\hat{s}_i$ can be approximated using the equation:

$$\hat{s}_i = c_0 + \sum_{j \neq i} c_j g\left(cos(\theta_{i,j})\right)$$

Where $\theta_{i,j}$ is the angle between the electrode locations i and j. Instead of calculating the angle, given the positions of the electrodes in space $p_i = (x_i, y_i, z_i)$ it is possible to directly calculate the cosine value: $\cos(\theta_{i,j}) = \frac{x_i \cdot x_j}{\|x_i\| \cdot \|x_j\|}$. The function $g(x)$ is defined as the sum of the series:

$$g(x) = \frac{1}{4\pi} \sum_{n=1}^{\infty} \frac{2n+1}{n^m (n+1)^m} P_n(x)$$

Where P_n is the Legendre polynomial and following [10] we set $m = 4$. Additionally, following the *eeglab* implementation we limited the infinite sum to values between 1 and 7. Finally, the coefficients $C = (c_1, c_2, \ldots, c_n)$ are set to be the solution for the system of equation $GC + Tc_0 = S$ with the constraint $T'C = 0$ where $G_{k,l} = g\left(cos(\theta_{k,l})\right)$, $S = (s_1, s_2, .., s_n)$ and T is a vector of ones. The channel being interpolated is excluded from the calculations of G and S.

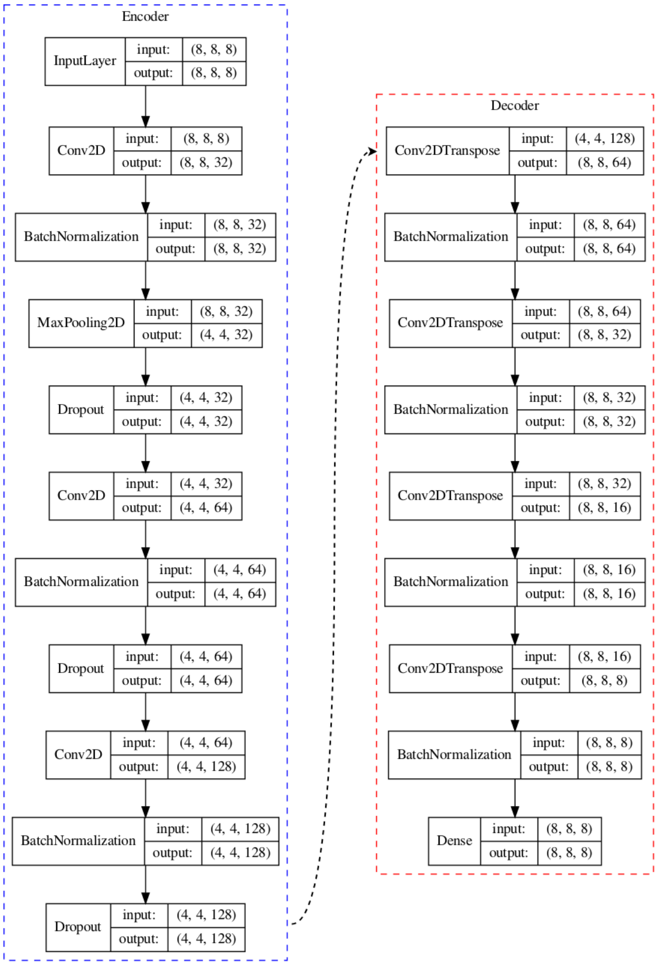

Fig. 3. Our network, the dashed black arrow denotes the $4 \times 4 \times 128$ embedded data tensor. This embedded representation is passed form the encoder to the decoder which then reconstructs the original input sans the occlusion.

D. Model Evaluation Approach

The selected baseline approaches [13] used the averaged normalized mean square error (ANMSE) as the main evaluation measure. The normalization of the mean square error is used to prevent a specific channel's performance from skewing the results, in case of a bad reconstruction. Having Z-scored the data however all channels are guaranteed to have the same mean amplitude. Therefore we do not normalize our mean square error results. Hence our final evaluation is:

$$AMSE = \frac{1}{M} \sum_{j=1}^{M} \left(\frac{\sum_{i=1}^{N} (s_i - \hat{s}_i)^2}{N} \right)_j$$

Where N is the number of channels (19 in our specific case). M is the number of samples. Note that the expression for mean reconstruction error (the inner average) changes for every sample. s_i and $\hat{s}_i$ are as previously defined. This measure was used both for optimizing the power parameter for the different baselines (see Section IV-C) and calculating the final results presented momentarily.

TABLE I

Comparison between Encoder-decoder model and baselines using Averaged mean square error (AMSE); *lower is better*. Note that the transfer learning (last row) was trained using a different data set on each column. There was no transfer learning for the *Seen Task, Seen Subject* partition as this is the original data used to train the model. SS: spherical splines baseline EGL: geodesic length calculation; EUD: euclidean baseline.

Interpolation Method	Seen Task, Seen Subjects	Seen Task, Unseen Subjects	Unseen Tasks, Seen Subjects	Unseen Task, Unseen Subjects
SS Baseline	0.728	0.694	0.8238	0.779
EUD Baseline	0.5215	0.561	0.665	0.566
EGL Baseline	0.585	0.501	0.566	0.622
Our Encoder-Decoder Model	**0.452**	0.451	0.466	0.469
Our Encoder-Decoder Model + Transfer Learning	—	**0.449**	**0.453**	**0.461**

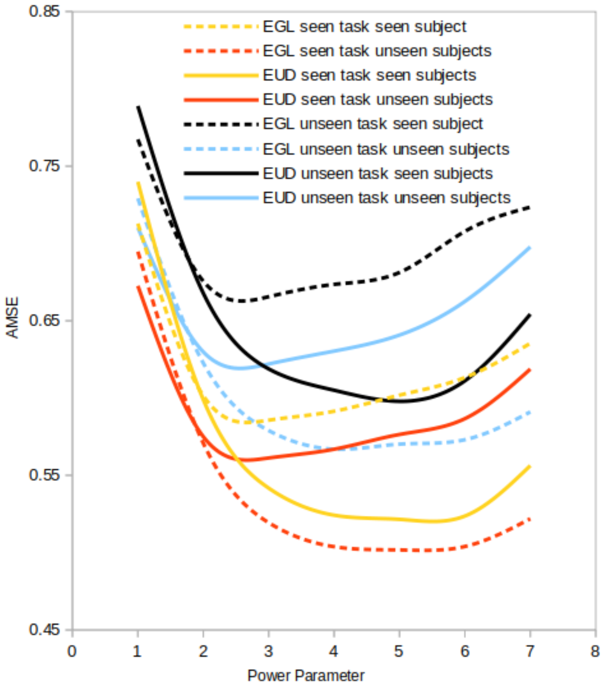

Fig. 4. Power parameter optimization to maximize the performance of the baseline approaches. For each of the four data partitions and for the two, EUD and EGL (solid and dashed lines respectively), methods. EGL: Geodesic Length calculation; EUD: euclidean baseline; AMSE: Averaged mean square error

V. Results

A. Model Hyper-parameter Optimization

After exploring the topological space by testing different network architectures (using the *Seen task, Seen subjects* data, see Figure 1), the best performing network is visualized in Figure 3. This network consisted of a simple encoder with three convolution layers and one max pooling layer, as well as four transposed convolutions in the decoder. Additionally there was a dropout and a batch normalization layer after each convolution in the encoder. The results described in this section were all achieved using this particular neural network architecture.

B. Baseline Power-parameter Optimization

For our evaluation to be extra rigorous. we optimized the power parameter for each baseline data-partition configuration separately. In Figure 4, we illustrate the results of our power parameter optimization for the baseline methods. As seen in the Figure, the optimal power parameters were comparable with those reported in previous literature (between 2 and 5) [13]. All the results that are reported in this section were for the optimized baseline on the specific data set being discussed. The spherical splines baseline has no analogous parameter we can optimize.

C. Main Result

In Table I, we compare the results of our proposed approach against the baselines for the EEG interpolation task on the test sets. The baseline methods are highly unstable, giving a high variability in performance relative to our approach.

The Encoder-decoder model consistently outperformed the baselines by at least 10%. Moreover, by utilizing transfer learning the network was able to improve it's accuracy even with minimal additional data and training time.

Interestingly, in contrast to results reported in the literature, the *EGL* method did not clearly outperform the *EUD* baseline [13]. This might be due to our data not having the precise electrode locations (see Subsection VI-B in the discussion).

Another interesting result is the pattern of improvements after transfer learning. As can be seen in the last row of Table I, the smallest improvement was for *Seen task, Unseen subjects* data. This discrepancy might hint that there was more variability between EEG data from different tasks compared to data from different subjects. Additional testing will be needed to verify this hypothesis. However, this can be seen as a compelling argument in favor of flexible models that can be tuned for the specific database the researcher is working with.

D. Performance on Exemplary Data

In Figure 5, we present an example of our method's interpolation on an exemplary portion of the data, compared against the baselines. As shown in the Figure, the reconstructions from other methods contain voltage fluctuations that do not appear in the original signal, or the one reconstructed using our method. This is evidence that our framework was able to learn the nuanced relationships between the electrode measurements that are not captured by the baseline approaches.

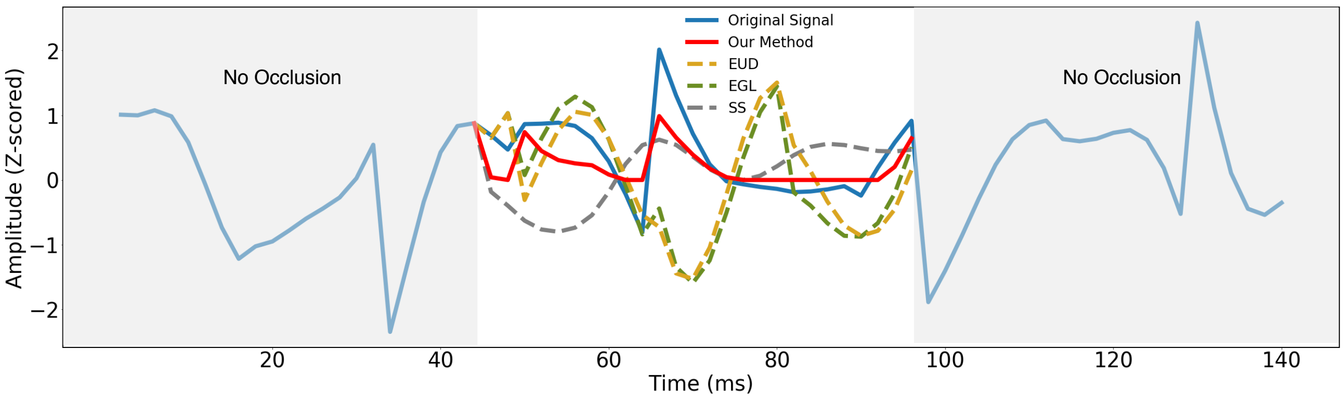

Fig. 5. An exemplary 48ms reconstruction of the EEG data for Subject 0 *resting state task* for channel *P4*. The original channel data was removed and interpolated using the three baselines and our method. EGL: geodesic length calculation; EUD: euclidean baseline; SS: spherical splines method.

VI. Discussion

Our work used a deep encoder-decoder model to tackle the problem of EEG channel interpolation. While discriminative frameworks are able to only detect and label bad data segments, our results demonstrate that a generative approach can reconstruct the missing channel with high fidelity to the original signal. The success of our method suggests that deep learning can capture complex relationships between electrodes that are not sufficiently expressed by the relatively simple inverse distance calculations that are the predominant contemporary solution.

A. On Self-supervised Learning

Data labeling is often a tenuous and resource consuming process. Unfortunately, training deep learning models often requires extensive data collection and labeling efforts. Therefore, deep learning researchers have recently began to focus on finding ways to mitigate the need for labeled data. As we showed in this study, one approach to mitigate this is to frame problems as a self-supervised learning tasks[2]. Our work is a special case of a popular self-supervised learning task: the prediction of occluded parts of data from visible ones. By using this framing we were able to circumvent a major challenge faced by deep learning approaches regardless of the field.

B. On the Challenges of Electrode Localization

As discussed previously, prior research that compared different interpolation methods used electrode localization to extract exact channel locations for each specific subject. While generic and imprecise locations are always available, electrode localization methods attempt to alleviate the noisiness inherent to EEG by providing exact locations of EEG electrodes. This localization can be done in many ways; one expensive option is to equip EEG caps with spatial sensors, or motion capture sensors [13], [25]. Other methods that require less specialized hardware including a simple DSLR camera [26] and Kinect with an Neural Network [27]. However, despite these recent advances, electrode localization remains uncommon. For instance, no EEG data set in physionet[3] or gigadb[4] contain an EEG database with electrode localization. Therefore, and to ensure our method is applicable to the vast majority of databases, the data we used also did not include electrode localization [18]. A possible future work could incorporate location data into the deep learning framework.

C. On Baseline Approaches

It is worth noting that there are multiple other interpolation methods such as the nearest neighbors method, planar-spline technique [14]. We selected the baselines methods described in Subsection II-A as they were the most contemporary approaches on the topic. Furthermore, the performance improvement of our model are especially impressive considering that the *EUD* and *EGL* baselines were optimized to maximize their performance on each and every separate partition of the data.

The *SS* method requires a system of equations to be solved for each and every time point. This is not a trivial requirement as it necessities complex calculations. This demand renders the *SS* method ill-suited for any online interpolation, and by extension many BCI applications [5], [28], [2]. In contrast to the taxing nature of the training procedure, piping data foreword in neural network is computationally cheap. Therefore our approach could potentially satisfy a growing need for accurate interpolation from online data.

D. On Transfer Learning

Transfer learning involves training a model on a problem similar to the one being solved. This is especially useful when only scarce data is available for the problem being solved, hindering the training of the model. While transfer learning is possible for many machine learning algorithms such as Bayesian networks and Markov chains, this technique

[2] For a particularly interesting primer on self-supervised learning see Yann LeCun's Keynote lecture at the Thirty-Forth annual meeting of AAAI.

[3] `https://physionet.org/about/database/#ecg`

[4] `http://gigadb.org/search/new?keyword=eeg`

became essential to deep learning especially due to its reliance on huge amounts of training data. Transfer learning is considered to be essential for the success and ubiquity of neural networks [29]. Our work for instance would be considerably less useful if it required every researcher to train the neural network from scratch, or if the results on data-sets that the model was not trained on were considerably worse.

VII. Conclusion and Future Work

With the increasing prevalence of EEG devices, there is a need for methodologies that better address common EEG artifacts. In this work, we developed a deep encoder-decoder based method to interpolate EEG segments impacted by the most common EEG artifact: the electrode *"pop"*. We demonstrated that our method improved EEG reconstruction performance compared to existing approaches, and that our method generalized well to unseen tasks and subjects.

Future work will extend this method to tackle other kinds of electrode artifacts. Moreover, an end-to-end system that automatically detects artifacts and replaces the corrupted data with an interpolated reconstruction of the original might be of particular interest to the community.

References

[1] V. S. Selvam and S. Shenbagadevi, "Brain tumor detection using scalp eeg with modified wavelet-ica and multi layer feed forward neural network," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6104–6109, 2011.

[2] X. Qian, B. Loo, F. Castellanos, S. Liu, H. Koh, X. Poh, R. Krishnan, D. Fung, M. Chee, C. Guan, T.-S. Lee, C. Lim, and J. Zhou, "Brain-computer-interface-based intervention re-normalizes brain functional network topology in children with attention deficit/hyperactivity disorder," *Translational Psychiatry*, vol. 8, p. 149, 08 2018.

[3] G. Vecchiato, F. Babiloni, L. Astolfi, J. Toppi, G. Vecchiato, L. Astolfi, P. Cherubino, J. Dai, W. Kong, and D. Wei, "Enhance of theta eeg spectral activity related to the memorization of commercial advertisings in chinese and italian subjects," in *2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)*, vol. 3, pp. 1491–1494, 2011.

[4] V. Siripornpanich, K. Sampoon, S. Chaithirayanon, N. Kotchabhakdi, and N. Chutabhakdikul, "Enhancing brain maturation through a mindfulness-based education in elementary school children: a quantitative eeg study," *Mindfulness*, vol. 9, 03 2018.

[5] L. Bonnet, F. Lotte, and A. Lécuyer, "Two brains, one game: Design and evaluation of a multiuser bci video game based on motor imagery," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 5, no. 2, pp. 185–198, 2013.

[6] W. Khalifa, A. Salem, M. Roushdy, and K. Revett, "A survey of eeg based user authentication schemes," in *2012 8th International Conference on Informatics and Systems (INFOS)*, pp. BIO–55–BIO–60, 2012.

[7] T. Walczak and S. Chokroverty, *Electroencephalography, Electromyography, and Electro-Oculography. General Principles and Basic Technology*, pp. 157–181. Elsevier Inc., 12 2009.

[8] E. K. S. Louis, L. C. Frey, J. W. Britton, J. L. Hopp, P. J. Korb, M. Z. Koubeissi, W. E. Lievens, and E. M. Pestana-Knight, "Electroencephalography (eeg): An introductory text and atlas of normal and abnormal findings in adults, children, and infants," 2016.

[9] Gabard-Durnam, M. L. L. J., W. A. S., C. L., and A. R. Levin, "The harvard automated processing pipeline for electroencephalography (happe): Standardized processing software for developmental and high-artifact data," *Frontiers in neuroscience*, vol. 12, 2018.

[10] F. Perrin, J. Pernier, O. Bertrand, and J. Echallier, "Spherical splines for scalp potential and current density mapping," *Electroencephalography and Clinical Neurophysiology*, vol. 72, no. 2, pp. 184 – 187, 1989.

[11] T. C. Ferree, "Spherical splines and average referencing in scalp electroencephalography," *Brain Topography*, vol. 19, pp. 43–52, Dec 2006.

[12] S. Petrichella, L. Vollere, F. Ferreri, A. Guerra, S. Määtta, M. Könönen, V. Di Lazzaro, and G. Iannello, "Channel interpolation in tms-eeg: A quantitative study towards an accurate topographical representation," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 989–992, Aug 2016.

[13] H. S. Courellis, J. R. Iversen, H. Poizner, and G. Cauwenberghs, "Eeg channel interpolation using ellipsoid geodesic length," in *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 540–543, Oct 2016.

[14] A. C. Soong, J. C. Lind, G. R. Shaw, and Z. J. Koles, "Systematic comparisons of interpolation techniques in topographic brain mapping," *Electroencephalography and Clinical Neurophysiology*, vol. 87, no. 4, pp. 185 – 195, 1993.

[15] M. Ghassemi, E. Amorim, T. Alhanai, J. Lee, S. Herman, A. Sivaraju, N. Gaspard, L. Hirsch, B. Scirica, M. Donnino, S. Biswal, V. Junior, S. Cash, E. Brown, R. Mark, and M. Westover, "Quantitative eeg trends predict recovery in hypoxic-ischemic encephalopathy," *Critical Care Medicine*, vol. 47, no. 10, pp. 1416–1423, 2019.

[16] B. McCartney, J. Martinez-del Rincon, B. Devereux, and B. Murphy, "A zero-shot learning approach to the development of brain-computer interfaces for image retrieval," *PLOS ONE*, vol. 14, pp. 1–21, 09 2019.

[17] P. Nejedly, J. Cimbalnik, P. Klimeš, F. Plesinger, J. Halamek, V. Křemen, I. Viscor, B. Brinkmann, M. Pail, M. Brazdil, G. Worrell, and P. Jurak, "Intracerebral eeg artifact identification using convolutional neural networks," *Neuroinformatics*, vol. 17, 08 2018.

[18] I. Zyma, S. Tukaiev, I. Seleznov, K. Kiyono, A. Popov, M. Chernykh, and O. Shpenkov, "Electroencephalograms during mental arithmetic task performance," *Data*, vol. 4, 01 2019.

[19] Z. Lin, M. Courbariaux, R. Memisevic, and Y. Bengio, "Neural networks with few multiplications," 10 2015.

[20] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, "Context encoders: Feature learning by inpainting," 2016.

[21] Y. Yang, X. Guo, J. Ma, L. Ma, and H. Ling, "Lafin: Generative landmark guided face inpainting," 2019.

[22] Y. Wang, X. Tao, X. Shen, and J. Jia, "Wide-context semantic image extrapolation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1399–1408, 2019.

[23] M. H. Baig, V. Koltun, and L. Torresani, "Learning to inpaint for image compression," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 1246–1255, Curran Associates, Inc., 2017.

[24] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research*, vol. 13, no. Feb, pp. 281–305, 2012.

[25] P. Reis and M. Lochmann, "Using a motion capture system for spatial localization of eeg electrodes.," *Frontiers in Neuroscience*, vol. 9, 04 2015.

[26] T. Clausner, S. S. Dalal, and M. Crespo-García, "Photogrammetry-based head digitization for rapid and accurate localization of eeg electrodes and meg fiducial markers using a single digital slr camera," *Frontiers in Neuroscience*, vol. 11, p. 264, 2017.

[27] N. Gessert, M. Gromniak, M. Bengs, L. Matthäus, and A. Schlaefer, "Towards deep learning-based eeg electrode detection using automatically generated labels," *Computer Vision and Pattern Recognition*, vol. abs/1908.04186, 2019.

[28] C. Lin, S. Tsai, and L. Ko, "Eeg-based learning system for online motion sickness level estimation in a dynamic vehicle environment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1689–1700, 2013.

[29] A. Ng, "Nuts and bolts of building ai applications using deep learning." NIPS Tutorial, 2016.