Deep Neural Imputation: A Framework for Recovering Incomplete Brain Recordings

Sabera Talukder[*]
Caltech
sabera@caltech.edu

Jennifer J. Sun[*]
Caltech
jjsun@caltech.edu

Matthew Leonard
UCSF
Matthew.Leonard@ucsf.edu

Bingni W. Brunton
U of Washington
bbrunton@uw.edu

Yisong Yue
Caltech
yyue@caltech.edu

Abstract

We study the problem of time series imputation in multivariate neural recordings. Compared to standard time series imputation settings, new challenges for imputing neural recordings include the lack of adjacent timestamps for electrodes missing over days, and generalization across days and participants with different electrode configurations. Due to these challenges, the standard practice in neuroscience is to discard electrodes with missing data, even if only a part of the recording is corrupted, significantly reducing the already limited and difficult-to-obtain data. In this paper, we establish Deep Neural Imputation (DNI), a framework to recover missing electrode recordings by learning across sessions, spatial locations, and participants. We first instantiate DNI with natural linear baselines, then develop encoder-decoder approaches based on masked electrode modeling. We evaluate DNI on 12 multielectrode, human neural datasets with naturalistic behavior. We demonstrate DNI's data imputation ability across a broad range of metrics as well as integrate DNI into an existing neural data analysis pipeline.

1 Introduction

Multielectrode recordings measure the dynamic activation of large networks of neurons in the brain and are a key enabling tool in studying neural function at scale [1, 2, 3, 4]. In humans, implanting such electrode arrays is an invasive neurosurgery procedure, and individual electrode implants can fail at any time. Therefore, there are almost always missing electrode values in these difficult-to-acquire and highly precious datasets. The current standard practice is to discard electrodes with missing data, excluding potentially useful information from analysis and limiting model generalizability across days and individuals [5, 2, 6]. To address this challenge, we study the problem of imputing missing values from multielectrode data using recordings collected across multiple days in 12 human datasets (Figure 1). While related to the generic problem of time-series imputation (e.g., [7, 8], more discussion in Section D), previous works are not directly applicable due to additional challenges associated with multielectrode recordings including: (1) electrodes are often missing for entire sessions with no adjacent recorded timestamps, and (2) existing methods do not handle data across participants who have completely different sets of electrodes.

Our contributions are summarized as follows:

[*] Equal contribution.

Workshop on Learning from Time Series for Health, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

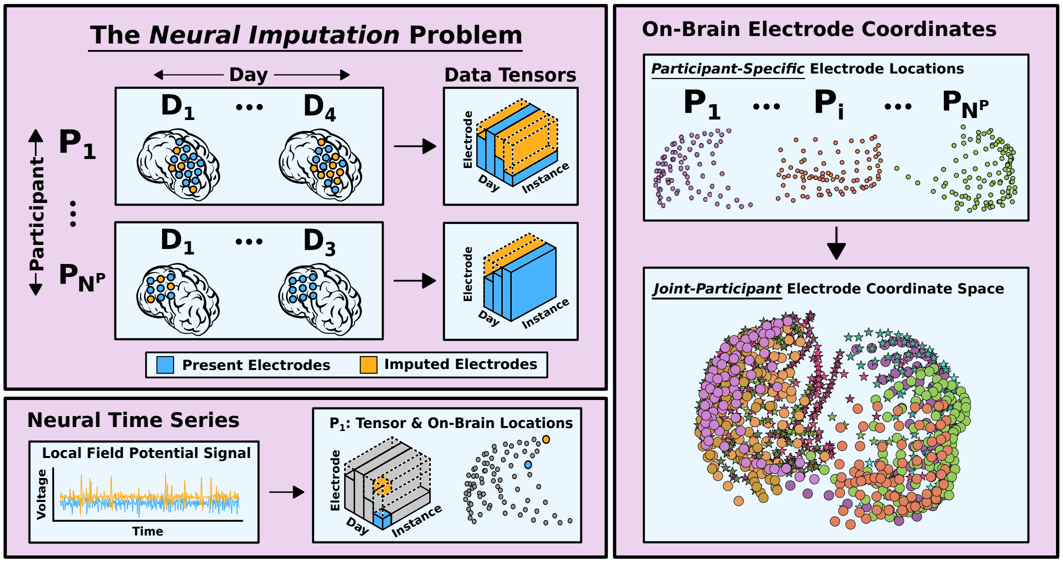
The Neural Imputation Problem
Day
D1 ••• D4
P1
Participant
⋮
PNP
D1 ••• D3
Data Tensors
Electrode
Day
Instance
Electrode
Day
Instance
Present Electrodes
Imputed Electrodes
Neural Time Series
Local Field Potential Signal
Voltage
Time
P1: Tensor & On-Brain Locations
Electrode
Day
Instance
On-Brain Electrode Coordinates
Participant-Specific Electrode Locations
P1 ••• Pi ••• PNP
Joint-Participant Electrode Coordinate Space

Figure 1: **Overview of Neural Imputation**. Neural imputation recovers missing electrode signals (yellow) across days and participants given non-corrupted data (blue). Each dataset contains data from a separate participant (N^P datasets total). *Top left:* Observed multielectrode data from each dataset/participant is a ragged tensor, because participants have different sets of electrodes that may be missing in different days. Note that electrodes may be missing for an entire session, which is atypical in standard time series imputation. In a typical neuroscience study, a few hours from a single day in a single participant are treated as an independent dataset for analysis (see [9, 2, 10], among many others). *Bottom left:* A time series instance from a single electrode in a specific brain location measures a voltage trace across time. *Right:* An additional challenge in imputation is variation in brain morphology and spatial electrode locations across participants.

- We propose the Deep Neural Imputation (DNI) Framework, a method for recovering fully missing multivariate electrode time series recordings across sessions and human participants. Additionally, we evaluate DNI's performance in 12 distinct human datasets where each participant performs naturalistic behavior with large inter-participant and inter-day variability [11, 12]. Canonically in neuroscience, only a few hours of a single participant's recordings would be treated as a distinct dataset and analyzed separately [9, 2, 10].
- We instantiate DNI using multiple linear and nonlinear imputation methods. The linear methods can be viewed as natural baseline approaches. The nonlinear DNI methods use a self-supervised task based on mask-filling during training, which we call masked electrode modeling. Masked electrode modeling is inspired by masking approaches in other domains [13, 7]. We then extend our learning approach to a multi-task setting to jointly model all participants across all 12 datasets.
- We provide experimental evidence of DNI's direct utility in downstream scientific analyses: (1) DNI reconstructs not only time series content but also frequency based power-spectral content (highly significant to the neuroscience community [14, 15, 16]), and (2) DNI's spatiotemporal reconstructions directly improve a brain decoder's classification accuracy when missing data is present.

2 The DNI Framework

In our problem formulation, participants are indexed by i, recording days are indexed by j, $\bar{\mathcal{E}}_{i,j}$ represents the set of observed electrodes on day j for participant i, and $\mathcal{E}_{i,j}$ represents the full set of active electrodes on day j for participant i (more discussion in Section A). The goal of DNI is to learn an imputation function $f : \bar{\mathcal{E}}_{i,j} \to \mathcal{E}_{i,j}$, as depicted in Figure 2 (Left). f may be either linear or nonlinear, and we study both instantiations. In the dataset-specific (participant-specific) case, we learn a function $f^{(i)}$ for each dataset (participant) i (Figure 2). We test these models on a held-out recording day, making them day-generalizable. In the across-dataset (joint-participant) case, we learn a single function f across all datasets (participants) (Figure 5). We call this model DOPE-generalizable,

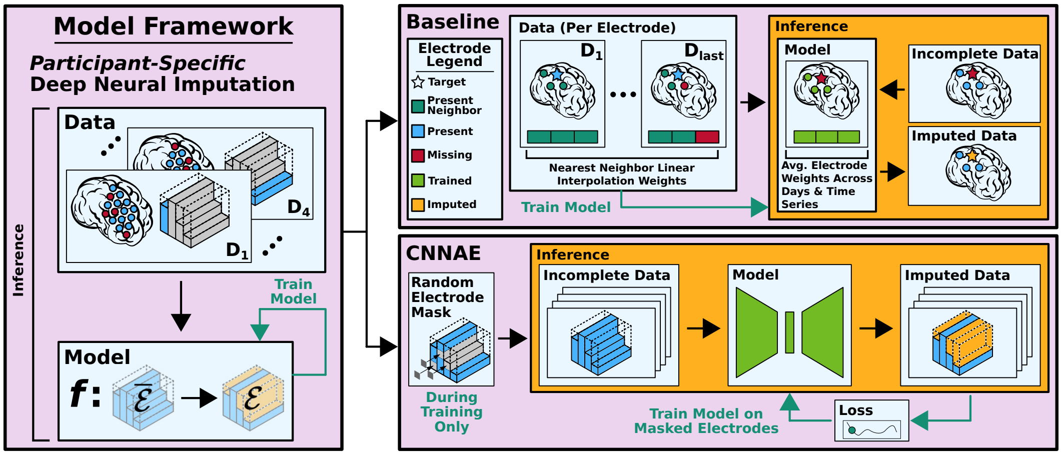

Figure 2: **DNI with Linear Nearest Neighbor Baseline and CNNAE Instantiations**. *Left:* In the participant-specific setting, we impute missing electrode recordings using data across days in the same participant. The goal is to map observed electrodes $\bar{\mathcal{E}}$ to the full set $\mathcal{E}$. *Top Right:* DNI with the linear NN baseline. *Bottom Right:* DNI with the CNNAE trained via masked electrode modeling.

Day-with-jOint-Participant-Embeddings-generalizable. By necessity, DOPE-generalizable models are also day-generalizable.

High-Level Approach. Since the neural imputation problem is not well-explored, a key design goal is to identify the simplest approach that works well for this domain. Two salient desiderata for imputation methods are (1) the recovery of missing data without adjacent timestamps using observations from different days; and (2) robustness to variations in brain morphology and physical electrode placement across participants. Day-generalizable participant-specific modeling addresses (1) but not (2), while DOPE-generalizable joint-participant modeling addresses both (1) and (2).

To create day-generalizable models that address (1), the key idea is to learn conserved relationships between electrodes across recording days. Conceptually straightforward methods are either the linear all electrode method or the linear nearest neighbors method, which we develop leveraging neuroscience domain knowledge. Either model is a natural baseline since no other baselines exist. We also study a nonlinear deep autoencoder model, Convolutional Neural Network AutoEncoder (CNNAE), that we later extend to train jointly over all participants, Multihead CNNAE (M-CNNAE). The M-CNNAE model is DOPE-generalizable and additionally addresses (2) (futher detail in Section B).

3 Experiments

We study DNI with a linear all electrode baseline model, linear nearest neighbors baseline model, CNNAE, and M-CNNAE on real-world multielectrode recordings from 12 human datasets where participants perform naturalistic behavior across multiple recording days. We summarize model comparisons (Section 3.1) and demonstrate that DNI recovers significant neural decoding performance with our imputations (Section 3.2). Additional details and results are presented in the Appendix, including dataset descriptions (C.1), training and evaluation setups (C.2), additional results (C.3), and frequency content evaluations (C.4), k-NN hyperparameter sweep (G.5), MSE metrics (G.6), and the linear all electrode performance(G.1).

3.1 Benchmark Results

Our models perform three categories of signal recovery: reconstruction (compared to observed signal), artificially induced imputation (we mask observed signal for evaluation), and natural imputation (the original signal was missing). We can perform direct quantitative evaluation of DNI's signal recovery capabilites with reconstruction and artificially induced imputation (i.e. imputation). Since there is no ground truth for naturally missing data, we do not evaluate natural imputation. Time series correlations comparisons between the linear NN, CNNAE, and M-CNNAE demonstrates that deep learning approaches generally have higher correlation (Figure 3). Prior approaches for imputing missing data in the neuroscience community use zero-filling, resulting in a correlation value of zero.

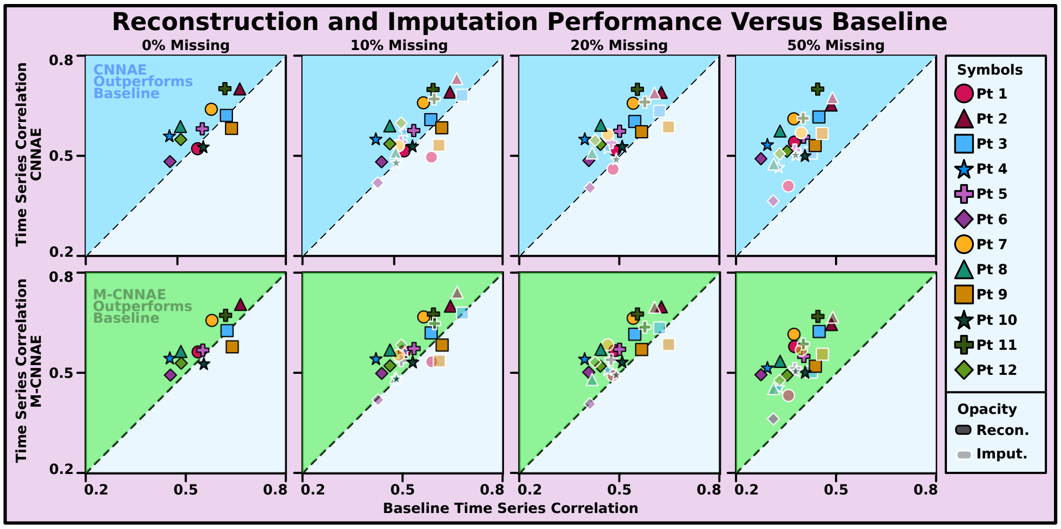

Figure 3: **Comparing Time Series Correlation Across DNI Methods**. We compare the CNNAE (top row) and M-CNNAE (bottom row) to the Linear Nearest Neighbor Baseline. Each point corresponds to a evaluation setting for one participant (mean of 9 runs - 3 runs of each model with 3 sets of missing electrodes).

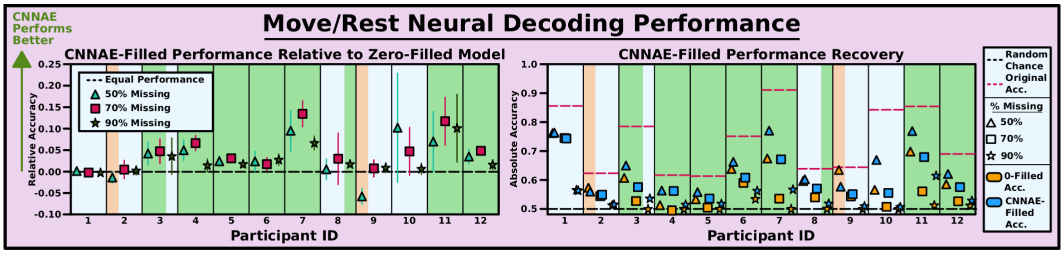

Figure 4: **Recovering Neural Decoding Performance When Missing Data is Present**. Background shading indicates (with 1 std. dev. of certainty) when the CNNAE-filled model outperforms (green), performs similarly to (light blue), or underperforms (yellow) compared to the zero-filled model. Each point corresponds to an experimental setting for one participant; results are averaged over 5 runs and the error bars represent 1 std. dev.

3.2 Downstream Task: Neural Imputation + Neural Decoding

Given neural decoding's significance in the neuroengineering community[4, 2, 3, 17], we study DNI's ability to directly improve neural decoding performance (Figure 4). We use the movement neural decoder from [18], which intakes neural time series data and predicts whether the neural data corresponds to either an arm movement event or rest. After recreating the random forest decoding performance for the original data, we randomly zero-filled either 50%, 70%, or 90% of the data for five random sets of missing electrodes and then computed the resulting performance. We purposefully chose these proportions to test DNI's limits on downstream applications with highly corrupted or missing data. Out of the 36 distinct experiments across 12 datasets, our CNNAE-filled data either increased or maintained decoding accuracy in 34/36 experiments (94.44% of the time). These results suggest that there is value in adopting DNI in the neuroscience and neuroengineering communities.

4 Conclusions

We presented DNI, a framework for imputing missing values in multivariate time series data tailored to the challenges in neuroscience data. By recovering previously unusable data, this line of work can amplify the strength of existing neuroscience analysis workflows. Future work includes pushing towards more powerful unified representations that can further enable joint-participant analysis.

References

[1] Angelique C Paulk, Yoav Kfir, Arjun R Khanna, Martina L Mustroph, Eric M Trautmann, Dan J Soper, Sergey D Stavisky, Marleen Welkenhuysen, Barundeb Dutta, Krishna V Shenoy, et al. Large-scale neural recordings with single neuron resolution using neuropixels probes in human cortex. Technical report, Nature Publishing Group, 2022.

[2] Gopala K Anumanchipalli, Josh Chartier, and Edward F Chang. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498, 2019.

[3] Francis R Willett, Donald T Avansino, Leigh R Hochberg, Jaimie M Henderson, and Krishna V Shenoy. High-performance brain-to-text communication via handwriting. *Nature*, 593(7858):249–254, 2021.

[4] Jose M Carmena, Mikhail A Lebedev, Roy E Crist, Joseph E O'Doherty, David M Santucci, Dragan F Dimitrov, Parag G Patil, Craig S Henriquez, Miguel A L Nicolelis, and Idan Segev. Learning to control a brain–machine interface for reaching and grasping by primates. *PLoS biology*, 1(2):e42, 2003.

[5] Kush Banga, Julius Benson, Niccolò Bonacchi, Sebastian A Bruijns, Rob Campbell, Gaëlle A Chapuis, Anne K Churchland, M Felicia Davatolhagh, Hyun Dong Lee, Mayo Faulkner, et al. Reproducibility of in-vivo electrophysiological measurements in mice. *bioRxiv*, 2022.

[6] Oliver Ruebel, Andrew Tritt, Benjamin Dichter, Thomas Braun, Nicholas Cain, Nathan Clack, Thomas Davidson, Max Dougherty, Jean-Christophe Fillion-Robin, Nile Graddis, et al. Nwb: N 2.0: an accessible data standard for neurophysiology. 2019.

[7] Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. Naomi: Non-autoregressive multiresolution sequence imputation. *Advances in neural information processing systems*, 32, 2019.

[8] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. Multivariate time series imputation with generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.

[9] Jennifer L Collinger, Brian Wodlinger, John E Downey, Wei Wang, Elizabeth C Tyler-Kabara, Douglas J Weber, Angus JC McMorland, Meel Velliste, Michael L Boninger, and Andrew B Schwartz. High-performance neuroprosthetic control by an individual with tetraplegia. *The Lancet*, 381(9866):557–564, 2013.

[10] Nikhilesh Natraj, Daniel B Silversmith, Edward F Chang, and Karunesh Ganguly. Compartmentalized dynamics within a common multi-area mesoscale manifold represent a repertoire of human hand movements. *Neuron*, 110(1):154–174, 2022.

[11] Steven M Peterson, Satpreet H Singh, Benjamin Dichter, Michael Scheid, Rajesh PN Rao, and Bingni W Brunton. Ajile12: Long-term naturalistic human intracranial neural recordings and pose. *Scientific Data*, 9(1):1–10, 2022.

[12] Dominic Gonschorek, Larissa Höfling, Klaudia P Szatko, Katrin Franke, Timm Schubert, Benjamin Dunn, Philipp Berens, David Klindt, and Thomas Euler. Removing inter-experimental variability from functional data in systems neuroscience. *Advances in Neural Information Processing Systems*, 34:3706–3719, 2021.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Kai J Miller, Eric C Leuthardt, Gerwin Schalk, Rajesh PN Rao, Nicholas R Anderson, Daniel W Moran, John W Miller, and Jeffrey G Ojemann. Spectral changes in cortical surface potentials during motor movement. *Journal of Neuroscience*, 27(9):2424–2432, 2007.

[15] Mainak Jas, Tom Dupré la Tour, Umut Simsekli, and Alexandre Gramfort. Learning the morphology of brain signals using alpha-stable convolutional sparse coding. *Advances in Neural Information Processing Systems*, 30, 2017.

[16] Scott Cole and Bradley Voytek. Cycle-by-cycle analysis of neural oscillations. *Journal of neurophysiology*, 122(2):849–861, 2019.

[17] Bingni W Brunton and Michael Beyeler. Data-driven models in human neuroscience and neuroengineering. *Current Opinion in Neurobiology*, 58:21–29, 2019.

[18] Steven M Peterson, Zoe Steine-Hanson, Nathan Davis, Rajesh PN Rao, and Bingni W Brunton. Generalized neural decoders for transfer learning across participants and recording modalities. *Journal of Neural Engineering*, 18(2):026014, 2021.

[19] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31, 2018.

[20] Steven M Peterson, Satpreet H Singh, Nancy XR Wang, Rajesh PN Rao, and Bingni W Brunton. Behavioral and neural variability of naturalistic arm movements. *Eneuro*, 8(3), 2021.

[21] Nancy Wang, Ali Farhadi, Rajesh Rao, and Bingni Brunton. Ajile movement prediction: Multimodal deep learning for natural human neural recordings and video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[22] Steven Michael Peterson, Rajesh PN Rao, and Bingni Wen Brunton. Learning neural decoders without labels using multiple data streams. *bioRxiv*, 2021.

[23] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[24] Joseph Young, Valentin Dragoi, and Behnaam Aazhang. Precise measurement of correlations between frequency coupling and visual task performance. *Scientific Reports*, 10(1):1–14, 2020.

[25] Rakesh Malladi, Don H Johnson, Giridhar P Kalamangalam, Nitin Tandon, and Behnaam Aazhang. Mutual information in frequency and its application to measure cross-frequency coupling in epilepsy. *IEEE Transactions on signal processing*, 66(11):3008–3023, 2018.

[26] Nicholas A Steinmetz, Christof Koch, Kenneth D Harris, and Matteo Carandini. Challenges and opportunities for large-scale electrophysiology with neuropixels probes. *Current opinion in neurobiology*, 50:92–100, 2018.

[27] John Lee, Max Dabagia, Eva Dyer, and Christopher Rozell. Hierarchical optimal transport for multimodal distribution alignment. *Advances in Neural Information Processing Systems*, 32, 2019.

[28] Chethan Pandarinath, K Cora Ames, Abigail A Russo, Ali Farshchian, Lee E Miller, Eva L Dyer, and Jonathan C Kao. Latent factors and dynamics in motor cortex and their application to brain–machine interfaces. *Journal of Neuroscience*, 38(44):9390–9401, 2018.

[29] Lorenzo Beretta and Alessandro Santaniello. Nearest neighbor imputation algorithms: a critical evaluation. *BMC medical informatics and decision making*, 16(3):197–208, 2016.

[30] Stef Van Buuren and Catharina GM Oudshoorn. Multivariate imputation by chained equations, 2000.

[31] Edgar Acuna and Caroline Rodriguez. The treatment of missing values and its effect on classifier accuracy. In *Classification, clustering, and data mining applications*, pages 639–647. Springer, 2004.

[32] Xiaoye Miao, Yangyang Wu, Jun Wang, Yunjun Gao, Xudong Mao, and Jianwei Yin. Generative semi-supervised learning for multivariate time series imputation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8983–8991, 2021.

[33] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

[34] William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: better text generation via filling in the_. *arXiv preprint arXiv:1801.07736*, 2018.

[35] Yonghong Luo, Ying Zhang, Xiangrui Cai, and Xiaojie Yuan. E2gan: End-to-end generative adversarial network for multivariate time series imputation. In *Proceedings of the 28th international joint conference on artificial intelligence*, pages 3094–3100. AAAI Press, 2019.

[36] Sander Dieleman, Charlie Nash, Jesse Engel, and Karen Simonyan. Variable-rate discrete representation learning. *arXiv preprint arXiv:2103.06089*, 2021.

[37] George Barnum, Sabera Talukder, and Yisong Yue. On the benefits of early fusion in multimodal representation learning. *arXiv preprint arXiv:2011.07191*, 2020.

[38] Nancy Xin Ru Wang. *Brains in the Wild: Machine learning for naturalistic, long-term neural and video recordings*. PhD thesis, 2018.

Appendix

A Background & Problem Setup

Data Modality & Format. Our data, further described in Section C.1, consists of participants (indexed by i), recording days (indexed by j),[2] electrodes corresponding to spatial locations (indexed by k), and time series instances (indexed by t). In other words, an observation $E_{i,j,k,t}$ corresponds to time series instance t from electrode k on day j for participant i. Indices that follow others demonstrate dependence; for example, time series instance t depends on the specific electrode k, the specific day j, and the specific participant i. Here we define generalizable as being able to perform on previously unseen data. When our models impute data across dimensions j, k, t they generalize across time series instances, spatial locations, and days (Figure 1) and we call them *day-generalizable*. When our models impute data across dimension i as well as dimensions j, k, t they additionally share a joint participant embedding space; we call them **D**ay-with-j**O**int-**P**articipant-**E**mbeddings-generalizable, or *DOPE-generalizable*. By necessity, DOPE-generalizable models are also day-generalizable.

Challenges in Modeling Electrode Recordings. Due to significant inter-participant and inter-day variability, data analysis methods in neuroscience commonly treat a few hours from single participants as individual datasets [9, 2, 10] to be analyzed separately. Figure 1's right panel demonstrates the stark electrode configuration variability in our participants, illustrating a common phenomenon in this data type. Further sources of this variability include movement artifacts, insurmountable experimental noise, and electrode recording failure. For example, in a neuropixel single-unit electrophysiology mouse study, only 55.4% of the data could be used for downstream analyses, because of "recording failure[s]," "low yield," and "noise/artifact[s]" [5]. In this ECoG LFP human speech decoding study, electrode recordings were discarded from a majority of the participants due to "bad signal quality" [2]. The pervasiveness of missing data is exemplified by the standard data storage file format in neuroscience, Neurodata Without Borders, which explicitly emphasizes its support for "dense ragged arrays" that permit "missing fields" [6].

These ragged data tensors result from the lack of consistency across the dimensions i, j, k, t (Top Left Figure 1). Since ragged data are difficult to jointly analyze, they are most commonly broken up into smaller, non-ragged tensors. This procedure leads to more manageable data for neuroscience analysis pipelines, meaning that discarding data remains a common practice despite its inefficiency. Because neuroscience data processing pipelines seldom handle variation in the expected data structure, these pipelines fundamentally limit the generalization capability of neuroscience models.

Formal Imputation Goals. To address the above challenges, our goal is to impute missing electrode values by learning from multielectrode time series data across spatial locations, days, and participants.

Let $\mathcal{E}_{i,j} = \{E_{i,j,k} \mid k \in \mathcal{K}_i\}$ be the full set of active electrodes on day j for participant i. $\mathcal{K}_i$ represents the full set of electrodes for participant i, which remains consistent across days. Let $\mathcal{M}_{i,j} \subset \mathcal{K}_i$ represent the set of missing electrodes on day j for participant i. Let $\bar{\mathcal{E}}_{i,j} = \{E_{i,j,k} \mid k \in \mathcal{K}_i \setminus \mathcal{M}_{i,j}\}$ represent the set of observed electrodes.

Formally, DNI seeks to recover $\mathcal{E}_{i,j}$ given $\bar{\mathcal{E}}_{i,j}$. Previous work for multivariate time series imputation [7, 19, 8] studies missing data at the level of sequences $\{t_1, t_2, ..., t_T\}$, where there may be missing observations for a set of timestamps (e.g. $\{t_5, t_6, t_7\}$). However, these methods could not impute when a feature was missing for all timestamps (e.g. $\{t_1, t_2, ..., t_T\}$); in other words, each feature required at least one observed timestamp. Therefore, these methods are not directly applicable to ours because an entire time series instance $E_{i,j,k,t} = \{t_1, t_2, ..., t_T\}$ may be missing and have no adjacent timestamps.

B The Deep Neural Imputation Framework - Additional Details

Linear All Electrode Baseline. We trained a linear all electrode model, with the details as follows: let n_e denote the total number of electrodes of a participant. The linear all electrode baseline learns a single set of weights $W \in \mathbb{R}^{n_e \times n_e}$ and biases $b \in \mathbb{R}^{n_e}$ for each participant. Suppose that $x \in \mathbb{R}^{n_e}$ represents the measured values (missing values are zeroed out) of all of a participant's electrodes at

[2] Most often, a neuroscience recording session is less than or equal to a few hours. However, for simplicity of exposition we use 24-hr days since that is how our data is organized.

some time step. Then, this baseline outputs $\hat{x} = Wx + b$. If $x_i = 0$, then $\hat{x}_i$ represents the imputed electrode value. We also set the diagonal values of W to be 0 since we cannot use an electrode's own value as a feature for imputation during inference. We train on MSE loss and models were trained until convergence by reconstructing present electrode values and were evaluated by imputing values we randomly zeroed out. The values we zeroed are the same across all previous models (i.e. NN-baseline, CNNAE, M-CNNAE) for comparison consistency.

Linear Nearest Neighbors Baseline. In the absence of appropriate existing baselines, we looked to known properties of neural signals for inspiration. Correlations exist across spatially close electrode neighbors; this well known observation motivated us to impute missing electrodes with linear combinations of observed, neighboring electrodes. In particular, we compute time series correlations from the N nearest neighbors of an electrode on observed days (e.g. "training days"). Then, we use these correlations as weights to linearly combine the time series from missing electrode neighbors on the held out "test day." These weighted, linearly combined time series compilations constitute our reconstructed neural signal and form our day-generalizable baseline model.

CNNAE. In our Convolutional Neural Network AutoEncoder (CNNAE) learning approach (Figure 2 Bottom Right), the encoder $f_\theta^{(i)}$ maps $\bar{\mathcal{E}}_{i,j}$ with zero-filled missing electrodes to an embedded representation $z_{i,j}$. The decoder $f_\psi^{(i)}$ then maps $z_{i,j}$ to $\hat{\mathcal{E}}_{i,j}$, which is either a reconstruction or an imputation (which we distinguish below) of the full set of electrodes $\mathcal{E}_{i,j}$. For $k \in \mathcal{K}_i \setminus \mathcal{M}_{i,j} = \overline{\mathcal{E}}_{i,j}$, $\hat{E}_{i,j,k}$ corresponds to a *reconstruction* of *observed* electrode data. For $k \in \mathcal{M}_{i,j}$, $\hat{E}_{i,j,k}$ corresponds to an *imputation* of *unobserved* electrode data. Imputation is a more challenging task than reconstruction, since in imputation, the decoding target is not an input to the model. The CNNAE is trained using the self-supervised objective described in Section B.1. Specific architecture design choices are discussed in Section C.2 – the key criterion is to effectively encode and decode multivariate electrode time series data. The CNNAE is a day-generalizable model, but it forms the backbone for the DOPE-generalizable model class described below.

M-CNNAE. For joint-participant imputation, the Multihead CNNAE model (M-CNNAE) learns a single function f for all participants, instead of one $f^{(i)}$ for each participant. When extending the CNNAE to jointly model different participants (Figure 5), we first used a participant-specific encoding layer to map input electrodes to a shared embedding space. This shared space is necessary because each participant has a different number of electrodes arranged in highly varied configurations. The encoder, f_θ, and decoder, f_ψ, are then trained using the CNNAE's learning procedure. To map the representation from the shared embedding space back to the participant-specific electrode configuration, we also train a participant-specific decoding layer. Components are trained jointly with data from all participants via a self-supervised objective (Section B.1). Unlike the CNNAE, the M-CNNAE trains a shared representation across all participants, making it DOPE-generalizable.

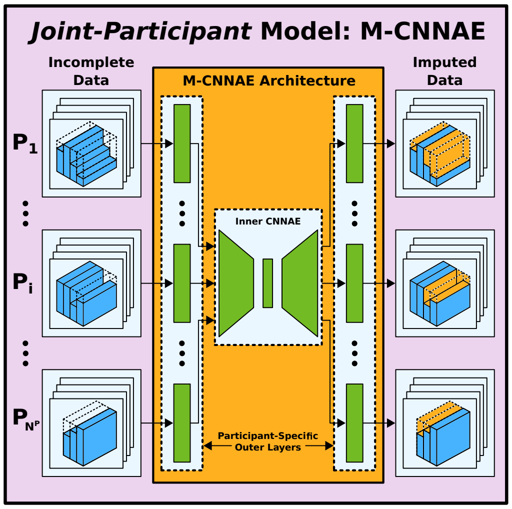

Figure 5: **Joint-Participant Deep Neural Imputation**. We extend our framework to jointly model many human participants, with different electrode spatial configurations, using participant-specific encoding and decoding layers and a shared CNNAE backbone.

B.1 Training Objective

We train our CNNAE and M-CNNAE models with self-supervision using masked electrode modeling, where we randomly mask l observed electrodes during training. Let $\tilde{\mathcal{E}}_{i,j}$ represent the observed electrodes with random masking. Then, we train the autoencoder to simultaneously impute the masked values as well as reconstruct the observed values. This task is similar to masking used in previous time series imputation approaches [7], except we mask entire time series from an electrode, instead of a subset of timestamps. The autoencoder is trained with the following objective: $\mathcal{L}_{i,j}^{NLL} = -\log\left(p_{f_\psi^{(i)}}\left(\bar{\mathcal{E}}_{i,j} | f_\theta^{(i)}(\tilde{\mathcal{E}}_{i,j})\right)\right)$.

C Experiments Continued

C.1 Datasets for Deep Neural Imputation

To study Deep Neural Imputation we utilize recently released data from all 12 AJILE12 [11] participants, each with $\sim$100 electrodes recorded across multiple days. Each participant has a wide range of naturalistic behaviors during recording, making reconstruction/imputation more difficult than if explored with canonical task-based neural data [20]. In the past, deep learning models have been used, often on subsets of the AJILE12 participants. However, these models have explored more classical neural decoding tasks, namely binary prediction and binary classification of arm movement [21, 18, 22].

Please see the table below for more details on each dataset, where each participant's data is one dataset. With each participant, days 1-2 are never used, as participants are still under heavy drug usage given their recent brain surgery. Days 6, 7, 7, 5-6 are discarded in participants 1, 2, 3, 5 respectively for other medical reasons. Days 3-7 are fully present in participants 4, 6, 7, 8, 9, 10, 11, 12. In the table, Pt is an abbreviation for participant.

Participant #	1	2	3	4	5	6	7	8	9	10	11	12
Total # of Days per Pt	4	4	4	5	3	5	5	5	5	5	5	5
Specific Recording Days	3-5, 7	3-6	3-6	3-7	3,4,7	3-7	3-7	3-7	3-7	3-7	3-7	3-7

Data Processing Pipelines. Using the same training split as [18] where the last day is held out as the test day, we define two different data processing pipelines:

- Procedure A: lower frequency content data used in Section C.3
- Procedure B: higher frequency content data used in Sections C.4, 3.2

We use Procedure A to study our models' ability to reconstruct time series data. We perform standardization on a 50,000 sample segment, corresponding to a $100sec$ length time series, on a per electrode, per day, per participant basis. We divide the $100sec$ into $20sec$ and $80sec$ sections, calculate the mean (μ_{20}) and standard deviation (σ_{20}) of the $20sec$ segment, and for each time step in the $80sec$ segment compute $\frac{TimeStep-\mu_{20}}{\sigma_{20}}$. As typically done in neuroscience studies without task labels, performing local standardization on each time segment allows us to account for neural data distribution changes such as spatial electrode shifts, neural drift, habituation to external stimuli, etc. We then downsample this data by a factor of 100, resulting in 400 time steps of data at a frequency of $5Hz$.

Using Procedure B, we study the performance of our imputation model with two downstream tasks: frequency content preservation, and a scientifically-relevant neural decoding task. We follow the data processing pipeline outlined in [18], consisting of band-pass filtering, downsampling, and trimming, resulting in $\sim$1000 time steps of data at a frequency of $250Hz$.

Our main goal with DNI is to reconstruct the neural time series; therefore, with Procedure A we aggressively downsampled to reduce the high temporal resolution of the original data. When using Procedure A, the frequency content of the reconstructions was recovered in the limited frequency bands available for analysis (i.e. lower frequency bands due to downsampling). The purpose of Procedure B was to (1) test our models in a regime that featured different time series lengths and frequency content in the training data, (2) verify the frequency content recovery property in higher frequency bands, (3) use our reconstructions in a downstream neural decoding task.

C.2 Training and Evaluation Setup

Our train and test split follows [18], where each participant's last day is held out as the test day. For the linear baseline model, we use 3 nearest neighbors to compute the correlation weights on the train days. We then apply these weights to the time series from the 3 nearest neighbors on the test day and combined them to reconstruct/impute the missing electrode's neural signal. For the CNNAE and M-CNNAE, we built the encoder with strided temporal convolutions and we based the decoder on [23]; together these comprise the shared backbone architecture (see Appendix for more modeling details). During the CNNAE and M-CNNAE training, we use our self-supervised training objective to perform masked electrode modeling. Masked electrode modeling (Figure 2) consists of creating a random electrode mask where 5-10$\%$ of the electrodes in each batch update are zero-filled. We then train the models to decode the full set of input electrodes; decoded electrode values are either reconstructed or imputed.

For evaluation, we use Pearson's Correlation between the original time series and reconstructed/imputed time series. We explore our model in 3 missing data regimes: 10% missing, 20% missing, and 50% missing. Within each missing data regime, we average our correlation results over all time series instances, electrodes, and 3 distinct randomly generated sets of missing electrodes. A crucial step in evaluating our models' decoding capabilities is comparing our reconstructions/imputations against the ground truth data that is unseen by the model. It is worth noting that the 12 participants' original data does have naturally missing electrodes. In these cases we cannot compare our imputations because ground truth does not exist. This motivated us to create zero-filled masked electrode modeling because it allows us to evaluate not only our decoded reconstructions, but also our decoded imputations.

C.3 Benchmark Results Continued

We find that Deep Neural Imputation's linear baseline performs moderately well, validating our intuition behind shared information between neighboring electrodes. As the percentage of missing electrodes increases, the baseline performance understandably falls because there are fewer neighboring electrodes available. At the same time, the performance of both the CNNAE and M-CNNAE holds across our missing data regimes and improves over the baseline. For 0% missing data the CNNAE outperforms the baseline for 9 participants, and in the most challenging evaluation regime (50% missing data) the CNNAE outperforms the baseline for all 12 participants in both reconstruction and imputation. The M-CNNAE model shares the same trend as the CNNAE when compared to the baseline, and in the most difficult evaluation regime (50% missing data) also outperforms the baseline for all 12 participants in both reconstruction and imputation.

When studying the differences between the M-CNNAE and CNNAE, we found that participant 1 had the greatest performance improvement when using the M-CNNAE. Despite the fact that there are 12 trained CNNAE models (one for each participant) and 1 jointly-trained M-CNNAE model, there are more cases where the M-CNNAE to baseline performance is better than CNNAE to baseline performance. To decipher the M-CNNAE and CNNAE differences we decided to explore the M-CNNAE joint-participant representation space and CNNAE's participant-specific representation spaces. For a similar comparison, we . For a similar comparison, we stacked all 12 of the CNNAE's participant-specific representation spaces and analyzed the concatenation. The M-CNNAE's joint-participant embedding had more samples mapped to a shared space, when compared to the CNNAE's concatenated embedding space which had far more participant-specific clusters. For a table of the results please see G.

C.4 Frequency Correlation Analysis

In Figure 6, we explore the relationship between frequency correlation and time series correlation across two proportions of missing data (0% & 50%) because of frequency content's significance to the neuroscience community [15, 16]. We observe a positive relationship between these correlations, despite the fact that our models were not trained to perform reconstruction or imputation in the frequency domain. In particular, we note that the points for both reconstruction (0% & 50%) and imputation (50%) form a curve (Figure 6 Left), suggesting that time series correlation is predictive of frequency correlation.

We additionally visualize the original time series and corresponding spectrogram as well as the CNNAE decoded time series and corresponding spectrogram for both a typical and performant example. A visual inspection shows that our CNNAE produces frequency reconstructions resembling the ground truth spectrograms even for low frequency correlation values. It is possible that alternative metrics for spectrogram evaluation, such as mutual information in frequency space [24] [25], could lead to stronger quantitative correlation metrics compared to the Pearson's Correlation metric that we used for evaluation. For a table of the results please see G.

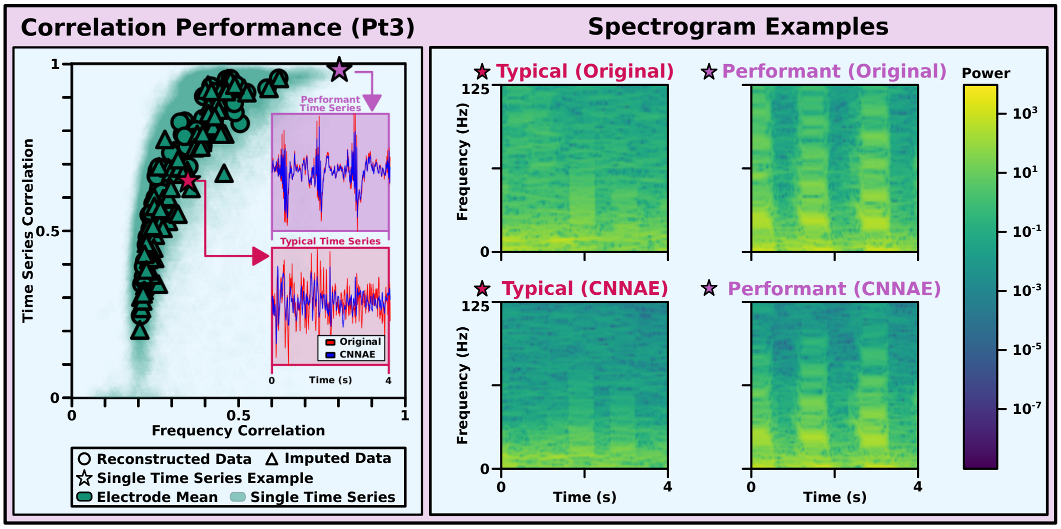

Figure 6: **Relationship between Frequency and Time Series Correlations**. *Left:* For participant 3 we plot the frequency vs. time series correlations along with two time series examples. Results for all patients can be found in the Appendix. *Right:* Spectrogram examples corresponding to the time series examples on the left for a typical example and a performant example.

D Related Work

Multielectrode Neuroscience Experiments. Elucidating scientific questions in neuroengineering, neural mechanism discovery, and systems neuroscience has been critically enabled by advances in multielectrode array technology. Different electrode recording modalities (EEG, ECoG, Utah arrays, Neuropixels [26, 1]) have specific trade-offs, such as varying signal attenuation or difficulty of implantation, to name a few. However, they all share the commonality that when individual electrodes are corrupted or fail from faulty manufacturing, electrical noise, scar tissue formation, etc., the signal is lost and cannot be experimentally recovered. Previous approaches address neural drift through alignment [27, 28], but not neural imputation. In this paper, we recover high-value incomplete brain recordings via our Deep Neural Imputation Framework.

Time Series Imputation. Our work relates to time series imputation, which aims to recover missing timestamps from time series data. In particular, multivariate time series modeling, where more than one feature is observed at each timestamp, has been studied using statistical methods [29, 30, 31] as well as deep generative machine learning models [7, 32, 19, 33]. Our linear model baseline falls under statistical approaches using nearest-neighbors, while our CNNAE and M-CNNAE approaches are based on generative modeling. A few common model setups include directly regressing missing values [19, 33], adversarial training (such as GANs) [32, 34, 8, 35], as well as autoencoders [7, 35]. Our CNNAE architecture is based on [36], a model recently developed for time series representation learning, which we adapted for time series imputation via masked electrode modeling. Methods for generic time series imputation typically cannot impute when a feature is missing for all timestamps. These methods are orthogonal to our main contribution, which is a framework for multivariate electrode imputation in both day-generalizable and DOPE-generalizable regimes.

E Discussion

In multielectrode recordings, signals from corrupted electrodes are commonly discarded and treated as missing. To fill these gaps, we propose Deep Neural Imputation: a framework to recover missing values using neural data across days and individuals. DNI is compatible with linear and nonlinear models, such as deep generative autoencoders, and can be easily be incorporated into existing neuroscience analysis pipelines and downstream tasks.

Key Observations. We find that our nearest neighbor linear baseline reconstructs and imputes missing electrode data well when there is a small percentage of data missing (e.g. 10%). Further, both the CNNAE and M-CNNAE models recover neural data in day-generalizable and DOPE-generalizable manners respectively when there are large percentages of data missing (e.g. 50%). These results may inspire neuroscientists to adjust neural electrode placement or enable novel experimental design. Moreover, there is a positive relationship between frequency correlation and time-series correlation, indicating that without training explicitly for frequency content, we can reconstruct and impute it. Finally, DNI is practical in downstream neuroscience tasks, which we validate based on significant improvement on a neural decoding task in the presence of missing data.

Limitations and Future Directions. There are many directions for future work stemming from the neural imputation problem. For joint-participant experiments, our method currently requires a participant-specific layer to be trained for each spatial configuration. Further explorations into transfer learning and meta-learning could eliminate this requirement. The joint-participant model offers an intriguing possibility of studying the joint embedding of neural representations to, for instance, understand individual variability in neural correlates of similar behaviors [20]. Additionally, our models cannot currently impute data gathered from unobserved spatial locations. Graph neural networks are one possible architecture that could develop this capability. Furthermore, these models may be improved by exploring multimodal fusion of neural, kinematic, and other measurement modalities [37, 22]. Since no prior work has conducted a rigorous study of how to recover missing electrode recordings across days and participants (see an early attempt [38]), establishing multi-participant, multi-day benchmark of DNI methods would be valuable.

Significance and Broader impacts. Participants' neural activity is deeply personal; therefore, we need to be thoughtful in our use of technology to analyze this information. The ethical considerations include issues of data management to guard the subjects' privacy and security. At the same time, we need to consider this technology's assistive potential in improving participants' lives, because recovering missing data from incomplete brain recordings has the potential to transform the design and analysis of a wide range of neuroscience and neuroengineering studies. Further discussion and information on the data collection process can be found in the original dataset release paper [11].

F Implementation Details

We provide additional details on the data processing and the model implementation. Model hyperparameters are in Table 1.

F.1 Data Processing

We will start by providing more intuition behind the type of neural time series that is collected from the participants. Each electrode in the ECoG array that participants are implanted with generates a local field potential (LFP) voltage trace. Each LFP trace comes from the bulk voltage activity of a few tens-of-thousands to hundreds-of-thousands of neurons in the brain. These LFP voltage traces the comprise the AJILE12 dataset are stored in the common neuroscience data file format, NWB, which is specifically designed to handle ragged data, missing values, and non-task-dependent (i.e. naturalistic) neural data [6].

In the AJILE12 dataset there are 12 human participants; each participant has roughly 100 ECoG electrodes implanted in their brain based on clinically determined locations. Before pre-processing, the data is collected at 500Hz and in most cases continuously recorded for several days. Given the multitude of goals in our paper we created two data processing pipelines to explore our models: Procedure A, and Procedure B. In Procedure A, we standardize 50,000 sample segments of the AJILE12 dataset and downsample the data to 5Hz to study our ability to perform time series reconstruction. In Procedure B, we following the data processing pipeline in [18], producing 250Hz data that allows us to study our frequency reconstruction ability, and our model's capability at restoring move/rest neural decoding performance.

F.2 Model Details

Linear Baseline Model. In our linear baseline, we first break the patients' recording days into "train" days and a "test" day. This data split follows the dataset split in [18], where the last day is held out as the test day. We then use the full set of electrodes for each patient to create a distance matrix which allows us to calculate each electrode's nearest neighbors by euclidean distance. The main idea is to linearly combine the time series correlation of the nearest neighbors to impute missing electrodes.

For every electrode on the test day, we compute the time series correlations between the target electrode and its three nearest neighbors averaged across observed training days. Then on the test day, the nearest neighbors are combined linearly using the train day correlations. Step-by-step, this corresponds to: (1) select a test day target electrode, (2) find one of its three nearest neighbors, (3) get the previously calculated time series correlation value that was averaged across all training days and all time series for that target electrode and the specific nearest neighbor, (4) take the time series from this nearest neighbor on the test day and weight the time series (multiply it) by the training time series correlation, (5) repeat 2-4 for all 3 nearest neighbors [if a neighbor is not observed on the test day, we use only the correlation from observed neighbors], (6) take the sum of the three weighted time series as the baseline reconstruction.

For evaluation, we (1) calculate the correlation between this baseline reconstruction and the time series of the actual target electrode recordings, (2) after calculating the correlations for all the time series off of an electrode we can average across all our time series to get the per electrode correlation.

CNNAE. The CNNAE model is trained for each participant. The model architecture is based on the encoder-decoder setup in [36], with strided temporal convolutions for encoding layers, an upsampling network, and a WaveNet decoder [23]. Our model hyperparameters are not tuned to AJILE12 and doing a hyperparameter search on the validation set can likely further improve performance. The model is trained to convergence of the train loss at epoch 100.

We input the time series data for the full set of electrodes for each participant (with missing electrodes zero-filled) to the model. At the input layer, we input both the electrode time series data as well as the time derivative of the data (change in time series value across 1 timestamp). The encoder downsamples the time series by a factor of 8 with the current kernel and stride configurations 1. We then upsample the encoding by a factor of 8 using the upsampling network before input to the decoder. The WaveNet decoder preserves the sequence length, and the output of WaveNet is fed into the final shallow temporal convolution layers to output the electrode reconstruction and imputation. There are four sets of shallow networks, outputting the mean and variance of the time series data and the time derivative. During evaluation, we use the output mean of the time series data.

Our CNNAE models are trained on either Amazon EC2 using p2 instances with a single Tesla K80 GPU, or a single NVIDIA GeForce RTX 2080Ti GPU. In addition to maximizing the log-likelihood of reconstruction and imputation during training, we also use the margin loss and slowness penalty proposed in [36] to regularize the embeddings.

M-CNNAE. The M-CNNAE model is trained jointly for all participants, with a similar architecture to the CNNAE. Our M-CNNAE model hyperparameters are not tuned to AJILE12 and doing a hyperparameter search on the validation set can likely further improve performance. The model is trained to epoch 45. The main difference for the M-CNNAE compared to the CNNAE is participant-specific encoding and decoding layers, with shared layers across participants similar to the CNNAE in between.

Given input time series data, the data is mapped by a participant-specific encoding layer before feeding into the CNNAE encoder, which is a set of strided convolutions. Then similar to the CNNAE, the encodings are upsampled, then fed into the WaveNet decoder. The decoder outputs are used as inputs to the final participant-specific decoding layers, which are shallow networks that map decoder outputs to the mean and variance of the time series data and the time derivative. During evaluation, we use the output mean of the time series data.

Our M-CNNAE models are trained on Amazon EC2 using p2 instances with a single Tesla K80 GPU. In addition to maximizing the log-likelihood of reconstruction and imputation during training, we also use the margin loss and slowness penalty proposed in [36] to regularize the embeddings.

G Additional Results

G.1 Linear All Electrode Results

See the table 2 for a comparison of the linear all electrode baseline, the nearest neighbors baseline, CNNAE, and M-CNNAE. Low MSE values are better. We find that the linear all electrode model performs worse than the 3-NN baseline across all conditions (10%, 20%, 50% missing) for participants

Model	Batch size	Learning Rate	z-dim	Upsampling	Num units	Encoder	Decoder
CNNAE	16	0.0001	64	8x	256	Kernel: [4,4,4] Stride: [2,2,2]	Layers: 2 Blocks: 2
M-CNNAE	2 (per pt)	0.0001	64	8x	256	Kernel: [4,4,4] Stride: [2,2,2]	Layers: 2 Blocks: 2

Table 1: **Hyperparameters for CNNAE and M-CNNAE.** The encoder is a set of strided convolutions with the specific kernel and stride parameters at each layer. The decoder is based on WaveNet [23]. The number of units is for both the encoder and decoder. The M-CNNAE has the same architecture as the CNNAE, except for shallow participant-specific encoding and decoding layers at the input and output respectively.

1, 2, 3, 5, 6, 8, 9, 10. For participants 4, 7, 11, 12 the learned linear baseline performance is most often similar to the 3-NN baseline.

G.2 Benchmark Results with Standard Deviation

The mean and standard deviation from the correlation comparison in Figure 3 of the main paper is shown in Tables 3, 4. At 0% missing electrodes, both the CNNAE and M-CNNAE generally outperform the baseline. In addition, the M-CNNAE performs better than the CNNAE for participant 1, and similarly to the CNNAE for the other participants. The gap between the machine learning models and the baseline increases as the amount of missing electrode increases - in particular, there is a significant improvement using the CNNAE and M-CNNAE across all participants at 50% of missing electrodes. Similar to Figure 3, the CNNAE and M-CNNAE generally has higher correlation compared to the baseline at imputation as well as reconstruction.

G.3 Frequency Analysis for All Participants

Here we expand on the participant 3 results presented in the paper, Figure 6, by showing the results for all participants. For participants 1, 2, 4 we show not only the time series and frequency correlation plots but also some example time series and spectrograms (Figures 7, 8, 9 respectively). In addition, for the other participants 5, 6, 7, 8, 9, 10, 11, 12 we show the time series and frequency correlation plots, Figure 10. Results for all patients are shown across two proportions of missing data, 10% and 50%.

We observed that there was a positive relationship between time series correlation and frequency correlation, despite not having trained our model on frequency reconstruction. This trend between time series correlation and frequency correlation can been seen strongest in participants 1, 2, 3, 6, 7, 8, 10, 11, and 12.

G.4 Neural Decoding Results

Table 5 expands on the results presented in Figure 4. In Table 5, we include the performance of the random forest neural decoder across 50%, 70%, and 90% of electrodes missing. We report the mean performance across 5 random seeds for the zero-filled data, and CNNAE-filled data. In addition, we include the mean and standard deviation for the pairwise relative accuracy between the CNNAE-filled data and zero-filled data. Positive mean relative accuracy values indicate that the CNNAE-filled data outperforms the zero-filled data on the move/rest neural decoding task.

G.5 k-Nearest Neighbors Hyperparameter Sweep

In the figures (11, 12) and tables (6, 7, 8, 9) below we see that the baseline performance plateaus as we add more neighbors across both conditions (0%, 10%, 20%, and 50% missing) as well as participants (1-12). This is in line with the brain's spatial pattern of correlation, a well-studied neuroscience phenomenon. The 3-Nearest Neighbors performance is very similar to the plateaued 7-Nearest Neighbors performance, as evidenced by Figure 12 (which is the Figure 3 recreation with 7NN baseline). The trend we observed in the main paper (3NN Linear Baseline - Figure 3) that our CNNAE or M-CNNAE model outperforms the linear baseline as the percent of missing data

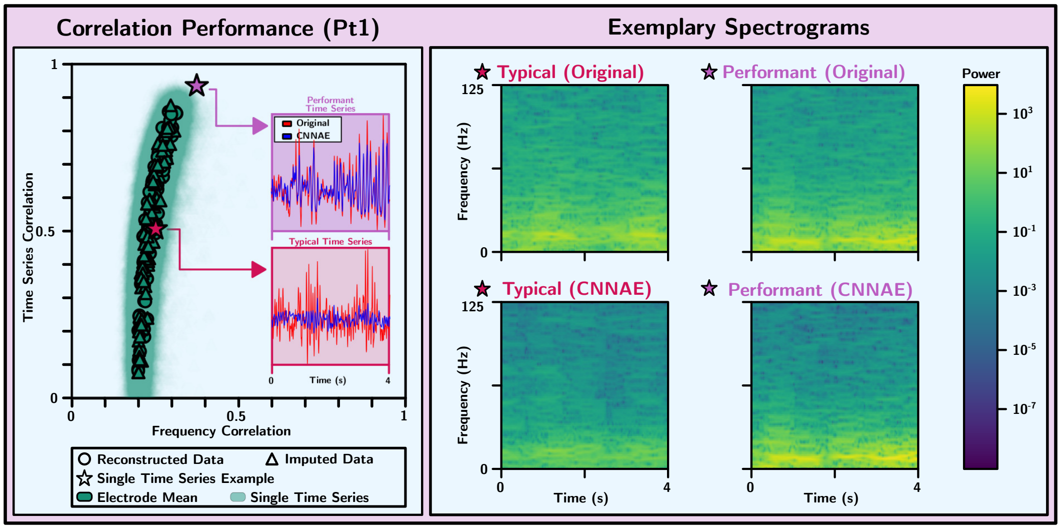

Figure 7: **Relationship between Frequency and Time Series Correlation**. *Left:* We show the frequency and time-series correlation from the CNNAE for participant 1, where each outlined point corresponds to one electrode. Two samples of reconstructed time series relative to the original are depicted. *Right:* Spectrogram examples corresponding to the time series examples on the left for a typical example and a performant example.

increases stays true with the 7NN Linear Baseline model (Figure 12). We want to emphasize that the state of the art in neuroscience is a correlation of 0 for imputation, because current practice discards fully-missing electrode time series. Therefore any of our linear baseline models (let alone our CNNAE and MCNNAE models) is a dramatic improvement above current practice.

G.6 Mean Squared Error (MSE) Metric

We compute the MSE as an additional metric to time series correlation. In general, we found that the models' rankings using MSE are similar to those using the correlation metric (Figure 3). In the table we present, lower MSE values are better. We also additionally report a Relative Imput. MSE value (which is the imputation minus the reconstruction value) so that one can directly compare reconstruction with imputation. Positive values mean that reconstructions outperform imputations. See tables 10 and 11 for details.

		% Electrodes Missing		
		10%	20%	50%
Pt 1	3-NN Baseline	.153 ± .007	.189 ± .010	.202 ± .010
	Learned Linear Baseline	.331 ± .028	.381 ± .062	.455 ± .076
	CNNAE	.173 ± .006	.184 ± .010	.210 ± .008
	M-CNNAE Baseline	.147 ± .005	.161 ± .012	.186 ± .007
Pt 2	3-NN Baseline	.531 ± .067	.502 ± .038	.502 ± .031
	Learned Linear Baseline	.977 ± .263	1.20 ± .284	1.59 ± .197
	CNNAE	.292 ± .028	.319 ± .027	.328 ± .010
	M-CNNAE Baseline	.283 ± .034	.314 ± .031	.351 ± .006
Pt 3	3-NN Baseline	.691 ± .393	.537 ± .212	.748 ± .319
	Learned Linear Baseline	.802 ± .113	.712 ± .050	.904 ± .228
	CNNAE	.212 ± .006	.233 ± .008	.542 ± .360
	M-CNNAE Baseline	.212 ± .007	.232 ± .009	.542 ± .355
Pt 4	3-NN Baseline	.308 ± .026	.306 ± .036	.338 ± .006
	Learned Linear Baseline	.225 ± .027	.247 ± .029	.313 ± .003
	CNNAE	.249 ± .016	.269 ± .030	.303 ± .004
	M-CNNAE Baseline	.244 ± .018	.269 ± .029	.301 ± .004
Pt 5	3-NN Baseline	3.54 ± 4.20	3.39 ± 2.28	2.68 ± .520
	Learned Linear Baseline	9.73 ± 5.27	7.61 ± 1.63	6.69 ± 1.78
	CNNAE	3.45 ± 4.18	2.72 ± 2.61	1.57 ± .966
	M-CNNAE Baseline	3.43 ± 4.16	2.69 ± 2.60	1.55 ± .964
Pt 6	3-NN Baseline	7.29 ± 3.06	7.00 ± 3.94	6.83 ± 1.10
	Learned Linear Baseline	10.6 ± 3.26	9.37 ± 2.75	7.98 ± .619
	CNNAE	3.93 ± 2.70	6.10 ± 4.57	6.29 ± 1.87
	M-CNNAE Baseline	3.94 ± 2.71	6.08 ± 4.57	6.27 ± 1.88
Pt 7	3-NN Baseline	.326 ± .035	.339 ± .017	.353 ± .014
	Learned Linear Baseline	.301 ± .006	.306 ± .020	.350 ± .006
	CNNAE	.312 ± .047	.305 ± .021	.302 ± .019
	M-CNNAE Baseline	.279 ± .048	.276 ± .014	.292 ± .011
Pt 8	3-NN Baseline	.386 ± .026	.411 ± .022	.441 ± .011
	Learned Linear Baseline	.490 ± .138	.431 ± .060	.475 ± .021
	CNNAE	.366 ± .021	.372 ± .018	.392 ± .008
	M-CNNAE Baseline	.376 ± .026	.384 ± .019	.398 ± .006
Pt 9	3-NN Baseline	.538 ± .060	.483 ± .060	.495 ± .044
	Learned Linear Baseline	.599 ± .065	.761 ± .055	.946 ± .103
	CNNAE	.401 ± .042	.364 ± .027	.403 ± .034
	M-CNNAE Baseline	.396 ± .038	.366 ± .024	.427 ± .035
Pt 10	3-NN Baseline	.433 ± .007	.582 ± .033	.410 ± .015
	Learned Linear Baseline	.557 ± .051	.801 ± .104	.940 ± .065
	CNNAE	.372 ± .022	.364 ± .015	.341 ± .014
	M-CNNAE Baseline	.362 ± .017	.393 ± .013	.338 ± .013
Pt 11	3-NN Baseline	.558 ± .024	.582 ± .030	.565 ± .022
	Learned Linear Baseline	.393 ± .049	.541 ± .073	.746 ± .114
	CNNAE	.385 ± .017	.364 ± .008	.394 ± .008
	M-CNNAE Baseline	.408 ± .013	.393 ± .008	.457 ± .015
Pt 12	3-NN Baseline	.354 ± .014	.370 ± .007	.388 ± .008
	Learned Linear Baseline	.246 ± .010	.292 ± .010	.389 ± .010
	CNNAE	.272 ± .019	.299 ± .006	.319 ± .005
	M-CNNAE Baseline	.280 ± .020	.307 ± .004	.337 ± .007

Table 2: **Imputation MSE Values for Learned Linear Baseline.** We compare imputation performance on the MSE metric between the NN baseline, new learned linear baseline (LLB), CNNAE, and M-CNNAE. Old performance values are copied from Table 10, 11 for convenience. The LLB performs worse than the 3-NN baseline across all conditions for Pt 1, 2, 3, 5, 6, 8, 9, 10. For Pt 4, 7, 11, 12 the LLB performance is most often similar to the 3-NN baseline. Low MSE values are better.

		% Electrodes Missing			
		0%	10%	20%	50%
Pt 1	Baseline	.536 -	.503 ± .007 .585 ± .010	.486 ± .012 .480 ± .023	.375 ± .027 .356 ± .020
	CNNAE	.521 ± .005 -	.516 ± .005 .497 ± .033	.519 ± .011 .461 ± .024	.542 ± .014 .411 ± .016
	M-CNNAE	.569 ± .008 -	.565 ± .009 .534 ± .034	.566 ± .015 .492 ± .021	.579 ± .019 .434 ± .023
Pt 2	Baseline	.662 -	.641 ± .006 .661 ± .067	.624 ± .014 .604 ± .045	.484 ± .031 .488 ± .031
	CNNAE	.700 ± .002 -	.691 ± .005 .731 ± .048	.690 ± .005 .688 ± .0029	.653 ± .013 .674 ± .014
	M-CNNAE	.710 ± .007 -	.700 ± .008 .741 ± .053	.697 ± .008 .696 ± .033	.645 ± .014 .667 ± .012
Pt 3	Baseline	.622 -	.583 ± .013 .677 ± .009	.545 ± .006 .619 ± .038	.448 ± .031 .425 ± .033
	CNNAE	.621 ± .001 -	.609 ± .003 .682 ± .010	.604 ± .003 .634 ± .019	.617 ± .018 .510 ± .022
	M-CNNAE	.632 ± .008 -	.620 ± .009 .679 ± .010	.615 ± .010 .634 ± .019	.624 ± .021 .503 ± .021
Pt 4	Baseline	.451 -	.419 ± .007 .505 ± .044	.396 ± .016 .464 ± .05	.294 ± .011 .327 ± .011
	CNNAE	.559 ± .003 -	.551 ± .007 .573 ± .035	.551 ± .013 .524 ± .046	.534 ± .014 .466 ± .009
	M-CNNAE	.550 ± .008 -	.543 ± .011 .565 ± .039	.541 ± .015 .509 ± .041	.516 ± .010 .458 ± .011
Pt 5	Baseline	.549 -	.532 ± .004 .494 ± .042	.500 ± .015 .475 ± .035	.402 ± .015 .376 ± .025
	CNNAE	.581 ± .012 -	.577 ± .014 .543 ± .035	.574 ± .016 .540 ± .035	.556 ± .021 .518 ± .017
	M-CNNAE	.573 ± .006 -	.573 ± .007 .538 ± .039	.570 ± .011 .539 ± .034	.550 ± .015 .514 ± .014
Pt 6	Baseline	.453 -	.436 ± .002 .425 ± .025	.408 ± .009 .411 ± .020	.274 ± .024 .311 ± .031
	CNNAE	.483 ± .013 -	.483 ± .017 .420 ± .019	.487 ± .016 .406 ± .029	.492 ± .019 .366 ± .016
	M-CNNAE	.499 ± .007 -	.500 ± .008 .421 ± .009	.502 ± .007 .408 ± .024	.495 ± .015 .364 ± .007

Table 3: **Correlation of Deep Neural Imputation Methods.** We show the time series correlation for the linear model baseline, CNNAE, and M-CNNAE. The top row for each method represents reconstruction, and the bottom row represents imputation. The mean and standard deviation is from 9 runs (3 runs of each model with 3 sets of missing data). Note that virtually all prior methods ignore missing data, therefore the conventional baseline approach would get 0 correlation. Participants 1 to 6 is shown, and due to space, participants 7 to 12 is in Table 4.

		% Electrodes Missing			
		0%	10%	20%	50%
Pt 7	Baseline	.577	.561 ± .007	.540 ± .015	.373 ± .032
		-	.487 ± .052	.465 ± .023	.395 ± .016
	CNNAE	.639 ± .016	.659 ± .007	.658 ± .006	.611 ± .016
		-	.531 ± .063	.564 ± .011	.570 ± .027
	M-CNNAE	.664 ± .010	.668 ± .012	.662 ± .011	.616 ± .015
		-	.553 ± .068	.584 ± .008	.569 ± .025
Pt 8	Baseline	.483	.461 ± .006	.444 ± .019	.332 ± .026
		-	.478 ± .024	.418 ± .047	.313 ± .045
	CNNAE	.588 ± .002	.590 ± .002	.592 ± .005	.575 ± .028
		-	.511 ± .016	.509 ± .016	.476 ± .005
	M-CNNAE	.570 ± .007	.569 ± .006	.569 ± .008	.535 ± .028
		-	.487 ± .025	.480 ± .016	.454 ± .005
Pt 9	Baseline	.637	.616 ± .006	.566 ± .011	.436 ± .038
		-	.608 ± .080	.646 ± .054	.457 ± .013
	CNNAE	.582 ± .001	.585 ± .006	.573 ± .009	.532 ± .016
		-	.533 ± .054	.587 ± .044	.568 ± .029
	M-CNNAE	.583 ± .006	.584 ± .008	.570 ± .012	.521 ± .011
		-	.537 ± .056	.584 ± .045	.557 ± .032
Pt 10	Baseline	.552	.529 ± .015	.508 ± .020	.406 ± .015
		-	.480 ± .110	.490 ± .076	.378 ± .014
	CNNAE	.526 ± .002	.529 ± .004	.529 ± .008	.501 ± .014
		-	.480 ± .020	.490 ± .018	.503 ± .019
	M-CNNAE	.532 ± .006	.533 ± .006	.532 ± .010	.502 ± .018
		-	.483 ± .024	.494 ± .019	.506 ± .020
Pt 11	Baseline	.617	.590 ± .004	.553 ± .007	.444 ± .074
		-	.594 ± .028	.575 ± .038	.401 ± .054
	CNNAE	.701 ± .001	.699 ± .005	.700 ± .005	.700 ± .024
		-	.671 ± .044	.661 ± .018	.613 ± .024
	M-CNNAE	.678 ± .009	.676 ± .011	.675 ± .011	.669 ± .024
		-	.648 ± .047	.636 ± .020	.587 ± .025
Pt 12	Baseline	.485	.461 ± .005	.442 ± .008	.352 ± .031
		-	.496 ± .031	.427 ± .015	.33 ± .023
	CNNAE	.549 ± .002	.536 ± .002	.535 ± .006	.515 ± .008
		-	.600 ± .027	.547 ± .018	.507 ± .009
	M-CNNAE	.535 ± .007	.522 ± .007	.520 ± .009	.494 ± .013
		-	.585 ± .027	.532 ± .019	.479 ± .011

Table 4: **Correlation of Deep Neural Imputation Methods.** We show the time series correlation for the linear model baseline, CNNAE, and M-CNNAE. The top row for each method represents reconstruction, and the bottom row represents imputation. The mean and standard deviation is from 9 runs (3 runs of each model with 3 sets of missing data). Note that virtually all prior methods ignore missing data, therefore the conventional baseline approach would get 0 correlation. Participants 7 to 12 is shown, and due to space, participants 1 to 6 is in Table 3.

		% Electrodes Missing		
		50%	70%	90%
Pt 1	Zero-Filled	0.7597	0.7430	0.5652
	CNNAE-Filled	0.7615	0.7412	0.5633
	Pairwise Relative Accuracy	0.0018 ± 0.0034	−0.0018 ± 0.0044	−0.0020 ± 0.0039
Pt 2	Zero-Filled	0.5726	0.5437	0.5126
	CNNAE-Filled	0.5593	0.5489	0.5156
	Pairwise Relative Accuracy	−0.0133 ± 0.0127	0.0052 ± 0.0217	0.0030 ± 0.0108
Pt 3	Zero-Filled	0.6062	0.5271	0.4997
	CNNAE-Filled	0.6487	0.5748	0.5359
	Pairwise Relative Accuracy	0.0425 ± 0.0266	0.0477 ± 0.0291	0.0363 ± 0.0426
Pt 4	Zero-Filled	0.5128	0.4948	0.5000
	CNNAE-Filled	0.5629	0.5614	0.5151
	Pairwise Relative Accuracy	0.0501 ± 0.0240	0.0667 ± 0.0181	0.0151 ± 0.0149
Pt 5	Zero-Filled	0.5331	0.5044	0.5000
	CNNAE-Filled	0.5576	0.5357	0.5181
	Pairwise Relative Accuracy	0.0245 ± 0.0130	0.0313 ± 0.0028	0.0181 ± 0.0088
Pt 6	Zero-Filled	0.6370	0.5893	0.5343
	CNNAE-Filled	0.6613	0.6070	0.5624
	Pairwise Relative Accuracy	0.0243 ± 0.0237	0.0178 ± 0.0156	0.0281 ± 0.0133
Pt 7	Zero-Filled	0.6728	0.5353	0.5000
	CNNAE-Filled	0.7682	0.6699	0.5669
	Pairwise Relative Accuracy	0.0954 ± 0.0477	0.1347 ± 0.0305	0.0669 ± 0.0163
Pt 8	Zero-Filled	0.5957	0.5395	0.5009
	CNNAE-Filled	0.6015	0.5699	0.5187
	Pairwise Relative Accuracy	0.0058 ± 0.0242	0.0303 ± 0.0600	0.0178 ± 0.0047
Pt 9	Zero-Filled	0.6333	0.5429	0.5000
	CNNAE-Filled	0.5762	0.5508	0.5095
	Pairwise Relative Accuracy	−0.0571 ± 0.0184	0.0079 ± 0.0201	0.0095 ± 0.0117
Pt 10	Zero-Filled	0.5656	0.5074	0.5000
	CNNAE-Filled	0.6680	0.5548	0.5069
	Pairwise Relative Accuracy	0.1024 ± 0.1269	0.0474 ± 0.0557	0.0069 ± 0.0131
Pt 11	Zero-Filled	0.6966	0.5601	0.5127
	CNNAE-Filled	0.7668	0.6777	0.6141
	Pairwise Relative Accuracy	0.0703 ± 0.0697	0.1176 ± 0.0559	0.1014 ± 0.0789
Pt 12	Zero-Filled	0.5845	0.5262	0.5120
	CNNAE-Filled	0.6199	0.5749	0.5286
	Pairwise Relative Accuracy	0.0354 ± 0.0165	0.0487 ± 0.0084	0.0166 ± 0.0120

Table 5: **CNNAE Recovers Neural Decoding Performance** (corresponding to Figure 6). Move/rest neural decoding performance absolute means, and pairwise relative accuracy means & standard deviations. The results for each percent missing (50%, 70%, and 90%) were calculated across 5 seeds. There are only two instances where the zero-filled accuracy outperformed the CNNAE-filled accuracy, corresponding to the yellow shading in Figure 6. All other cases correspond to where the CNNAE-filled accuracy was equal to or outperformed the zero-filled accuracy.

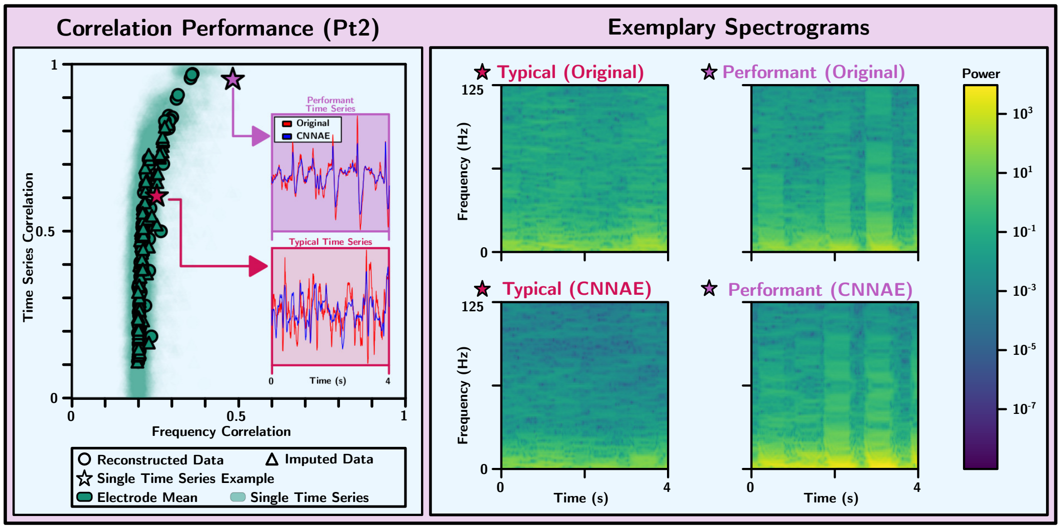

Figure 8: **Relationship between Frequency and Time Series Correlation**. *Left:* We show the frequency and time-series correlation from the CNNAE for participant 2, where each outlined point corresponds to one electrode. Two samples of reconstructed time series relative to the original are depicted. *Right:* Spectrogram examples corresponding to the time series examples on the left for a typical example and a performant example.

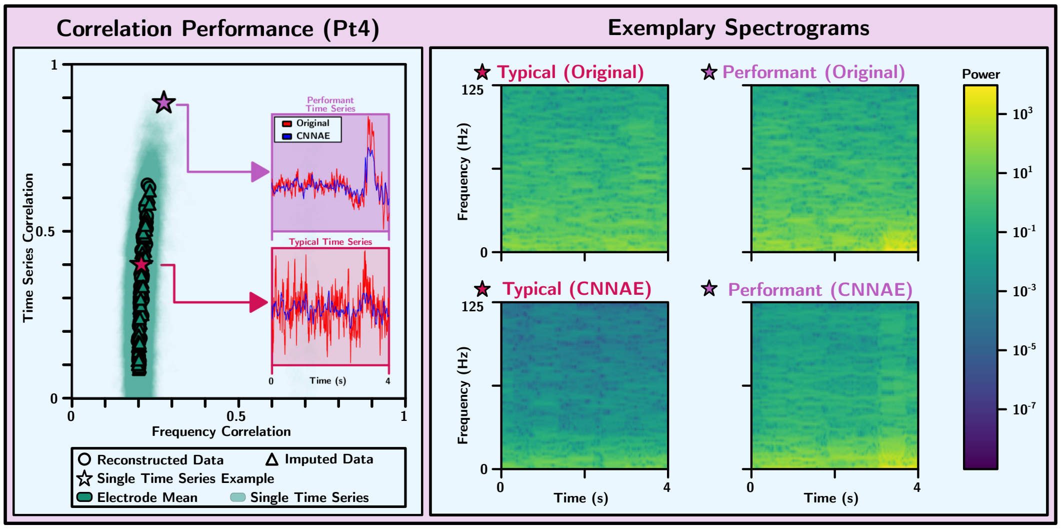

Figure 9: **Relationship between Frequency and Time Series Correlation**. *Left:* We show the frequency and time-series correlation from the CNNAE for participant 4, where each outlined point corresponds to one electrode. Two samples of reconstructed time series relative to the original are depicted. *Right:* Spectrogram examples corresponding to the time series examples on the left for a typical example and a performant example.

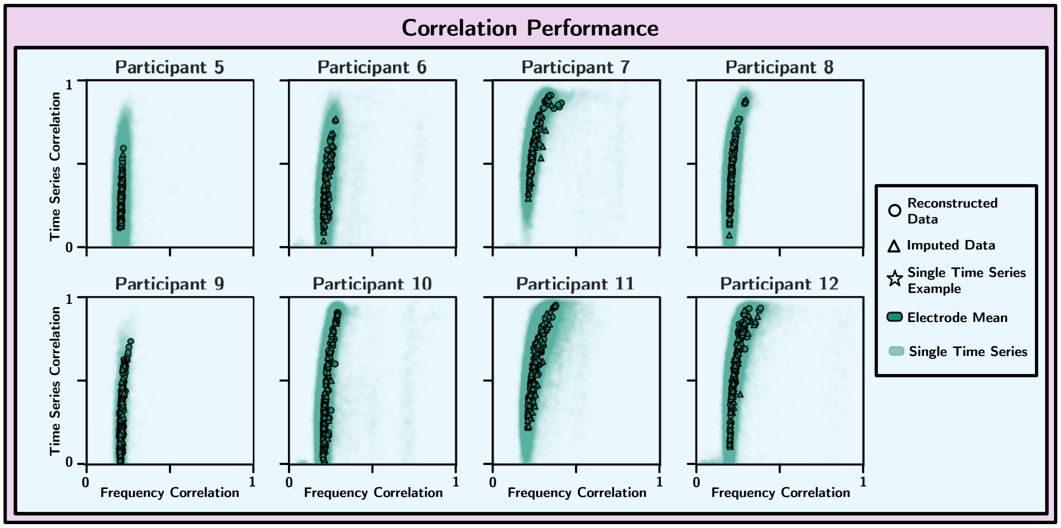

Figure 10: **Relationship between Frequency and Time Series Correlation**. *Left:* We show the frequency and time-series correlation from the CNNAE for participant 5, 6, 7, 8, 9, 10, 11, 12, where each outlined point corresponds to one electrode.

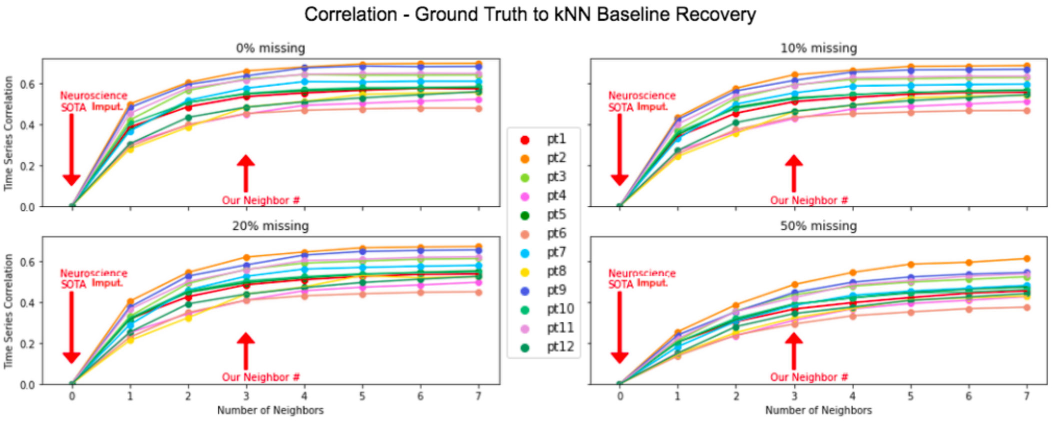

Figure 11: **Nearest Neighbor Hyperparameter Sweep**. We see that the baseline performance plateaus as we add more neighbors across both conditions (0%, 10%, 20%, and 50% missing) as well as participants (1-12). The 3NN linear baseline (model we use in Figure 3) is similar to the results for the plateaued 7NN model. Both of which are much better than the Neuroscience SOTA (State Of The Art) for imputation, which gives a 0 correlation. Here we report our time series correlation metric between our ground truth and reconstructed time series as a function of neighbor number. We average across time series instances, electrodes, days, and reconstruction/imputation conditions. Each participant is a different color. See tables below for the specific values presented in the figure.

Pt	Number of Nearest Neighbors							
	0	1	2	3	4	5	6	7
1	0	.386	.486	.536	.554	.567	.576	.575
2	0	.500	.604	.662	.681	.696	.698	.699
3	0	.421	.566	.622	.644	.641	.641	.642
4	0	.305	.396	.451	.493	.503	.514	.523
5	0	.372	.507	.549	.564	.573	.578	.581
6	0	.292	.399	.453	.469	.476	.480	.480
7	0	.367	.517	.577	.609	.608	.611	.611
8	0	.279	.386	.484	.512	.547	.553	.557
9	0	.482	.595	.637	.677	.686	.683	.684
10	0	.403	.507	.552	.571	.580	.581	.585
11	0	.454	.577	.617	.646	.648	.650	.649
12	0	.305	.434	.484	.510	.530	.545	.560

Table 6: k**-NN Hyperparameter Sweep - 0% Missing**. Time series correlations for the k-nearest neighbor baseline as a function of k for all 12 participants (higher is better). 0 neighbors (i.e. discarding data) is standard practice in the field of neuroscience, and always yields 0 correlation for imputation. Performance improves and quickly saturates as k increases.

Pt	Number of Nearest Neighbors							
	0	1	2	3	4	5	6	7
1	0	.340 ± .011	.455 ± .006	.511 ± .005	.532 ± .006	.548 ± .007	.556 ± .005	.557 ± .005
2	0	.435 ± 0.015	.577 ± .004	.643 ± .002	.665 ± .002	.684 ± .003	.686 ± .003	.688 ± .003
3	0	.371 ± .024	.528 ± .013	.593 ± .012	.620 ± .007	.623 ± .004	.628 ± .003	.630 ± .004
4	0	.271 ± .015	.364 ± .011	.428 ± .005	.475 ± .005	.488 ± .003	.500 ± .004	.511 ± .003
5	0	.350 ± .007	.484 ± .003	.529 ± .001	.546 ± .002	.556 ± .000	.564 ± .001	.568 ± .002
6	0	.256 ± .005	.374 ± .006	.435 ± .000	.452 ± .001	.461 ± .001	.467 ± .000	.468 ± .000
7	0	.331 ± .006	.499 ± .001	.554 ± .002	.588 ± .002	.592 ± .002	.595 ± .001	.597 ± .001
8	0	.243 ± .005	.356 ± .006	.463 ± .005	.493 ± .002	.530 ± .002	.537 ± .003	.543 ± .003
9	0	.424 ± .006	.564 ± .005	.615 ± .004	.656 ± .000	.667 ± .002	.668 ± .002	.670 ± .002
10	0	.362 ± .003	.478 ± .003	.524 ± .002	.546 ± .004	.557 ± .004	.560 ± .003	.565 ± .002
11	0	.402 ± .007	.539 ± .001	.591 ± .002	.628 ± .003	.632 ± .001	.636 ± .001	.636 ± .001
12	0	.272 ± .002	.409 ± .002	.464 ± .004	.494 ± .003	.516 ± .004	.531 ± .005	.545 ± .005

Table 7: k**-NN Hyperparameter Sweep - 10% Missing**. Mean time series correlations ±1 standard deviation for the k-nearest neighbor baseline as a function of k for all 12 participants (higher is better). 0 neighbors (i.e. discarding data) is standard practice in the field of neuroscience, and always yields 0 correlation for imputation. Performance improves and quickly saturates as k increases.

Pt	Number of Nearest Neighbors							
	0	1	2	3	4	5	6	7
1	0	.318 ± .005	.426 ± .010	.485 ± .014	.510 ± .013	.526 ± .014	.536 ± .011	.538 ± .009
2	0	.404 ± .014	.545 ± .011	.620 ± .011	.645 ± .003	.666 ± .005	.670 ± .004	.672 ± .003
3	0	.330 ± .019	.491 ± .015	.560 ± .006	.591 ± .003	.601 ± .007	.610 ± .002	.613 ± .002
4	0	.250 ± 0.015	.343 ± .007	.409 ± .006	.456 ± .007	.473 ± .006	.484 ± .003	.497 ± .002
5	0	.315 ± .012	.448 ± .008	.495 ± .005	.519 ± .003	.536 ± .004	.546 ± .006	.553 ± .004
6	0	.228 ± .011	.349 ± .008	.409 ± .007	.431 ± .006	.441 ± .006	.448 ± .004	.451 ± .002
7	0	.289 ± .017	.459 ± .009	.526 ± .008	.562 ± .005	.570 ± .001	.576 ± .003	.580 ± .003
8	0	.212 ± .001	.323 ± .006	.439 ± .005	.474 ± .004	.530 ± .002	.519 ± .003	.525 ± .002
9	0	.375 ± 0.005	.527 ± .009	.582 ± .003	.630 ± .008	.648 ± .002	.653 ± .003	.656 ± .003
10	0	.322 ± .005	.455 ± .004	.504 ± .004	.525 ± .009	.539 ± .007	.543 ± .006	.547 ± .006
11	0	.364 ± .010	.500 ± .002	.558 ± .006	.603 ± .008	.610 ± .005	.619 ± .002	.622 ± .002
12	0	.252 ± .004	.391 ± .006	.439 ± .005	.471 ± .002	.496 ± .003	.512 ± .004	.526 ± .007

Table 8: k**-NN Hyperparameter Sweep - 20% Missing**. Mean time series correlations ±1 standard deviation for the k-nearest neighbor baseline as a function of k for all 12 participants (higher is better). 0 neighbors (i.e. discarding data) is standard practice in the field of neuroscience, and always yields 0 correlation for imputation. Performance improves and quickly saturates as k increases.

Pt	Number of Nearest Neighbors							
	0	1	2	3	4	5	6	7
1	0	.205 ± .009	.304 ± .009	.366 ± .009	.397 ± .009	.422 ± .008	.444 ± .006	.455 ± .006
2	0	.255 ± .032	.387 ± .008	.486 ± .009	.545 ± .009	.586 ± .003	.595 ± .004	.613 ± .007
3	0	.212 ± .014	.357 ± .017	.436 ± .002	.478 ± .003	.498 ± .000	.511 ± .003	.524 ± .004
4	0	.152 ± .020	.236 ± .005	.311 ± .002	.367 ± .007	.393 ± .012	.409 ± .007	.427 ± .009
5	0	.204 ± .003	.313 ± .008	.389 ± .010	.421 ± .011	.448 ± .015	.464 ± .007	.475 ± .006
6	0	.137 ± .019	.239 ± .009	.293 ± .005	.333 ± .003	.352 ± .001	.368 ± .006	.375 ± .004
7	0	.182 ± .019	.306 ± .011	.384 ± .010	.433 ± .001	.454 ± .002	.468 ± .002	.482 ± .002
8	0	.145 ± .009	.251 ± .019	.322 ± .011	.369 ± .014	.411 ± .014	.421 ± .010	.430 ± .011
9	0	.239 ± .017	.353 ± .019	.447 ± .021	.497 ± .009	.523 ± .003	.537 ± .008	.545 ± .008
10	0	.204 ± .002	.322 ± .004	.392 ± .006	.420 ± .003	.444 ± .007	.450 ± .007	.463 ± .002
11	0	.222 ± .016	.354 ± .021	.422 ± .023	.485 ± .013	.505 ± .010	.527 ± .013	.538 ± .013
12	0	.150 ± .005	.280 ± .008	.344 ± .012	.376 ± .014	.408 ± .011	.425 ± .009	.442 ± .010

Table 9: k**-NN Hyperparameter Sweep - 50% Missing**. Mean time series correlations ±1 standard deviation for the k-nearest neighbor baseline as a function of k for all 12 participants (higher is better). 0 neighbors (i.e. discarding data) is standard practice in the field of neuroscience, and always yields 0 correlation for imputation. Performance improves and quickly saturates as k increases.

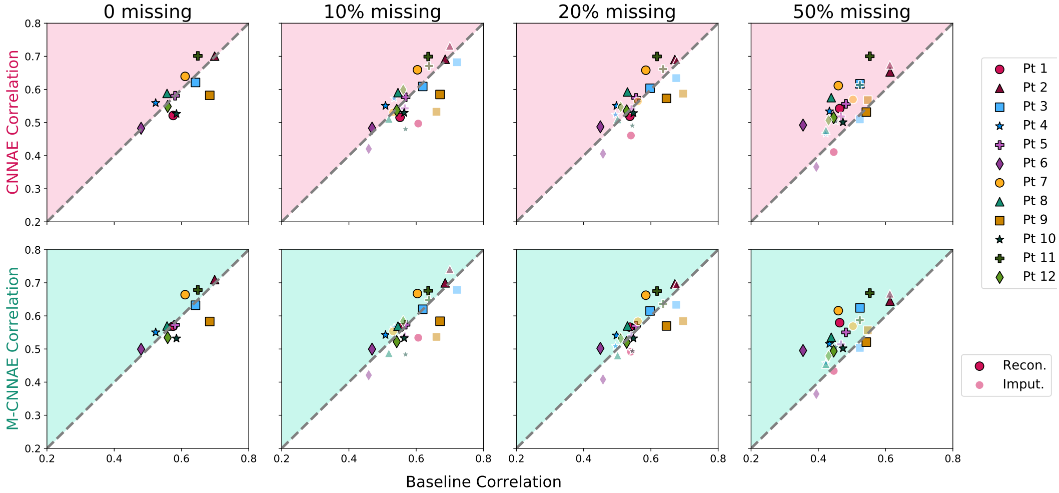

Figure 12: **Fig. 3 Using 7NN Linear Baseline.** The top row compares time series correlation for the 7NN linear model and CNNAE while the bottom row compares time series correlation for the 7NN linear model and M-CNNAE. The shaded triangle above the diagonal shows where the CNNAE and M-CNNAE outperforms the baseline. Each point corresponds to a evaluation setting for one participant; the mean over 9 runs are shown (3 runs of each model with 3 sets of missing data). Note that virtually all prior methods ignore missing data and thus conventional approaches correspond to 0 correlation. Observe that the trend of the CNNAE and M-CNNAE models outperforming the baseline with greater fractions of missing data holds even when we increase the number of nearest neighbors in the baseline model from 3 (Figure 3 in main paper) to 7 (this figure).

			% Electrodes Missing			
			0%	10%	20%	50%
Pt 1	Baseline	Recon. MSE	.188	.193 ± .002	.187 ± .002	.192 ± .003
		Imput. MSE	-	.153 ± .007	.189 ± .010	.202 ± .010
		Relative Imput. MSE	-	−0.040 ± .007	.002 ± .010	.010 ± .010
	CNNAE	Recon. MSE	.176 ± .000	.179 ± .001	.179 ± .002	.173 ± .006
		Imput. MSE	-	.173 ± .006	.184 ± .010	.210 ± .008
		Relative Imput. MSE	-	−.006 ± .006	.005 ± .010	.037 ± .010
	M-CNNAE	Recon. MSE	.145 ± .003	.148 ± .003	.149 ± .003	.148 ± .005
		Imput. MSE	-	.147 ± .005	.161 ± .012	.186 ± .007
		Relative Imput. MSE	-	−.001 ± .006	.012 ± .012	.038 ± .009
Pt 2	Baseline	Recon. MSE	.581	.539 ± .037	.536 ± .056	.511 ± .031
		Imput. MSE	-	.531 ± .067	.502 ± .038	.502 ± .031
		Relative Imput. MSE	-	−.008 ± .077	−.034 ± .068	−.009 ± .044
	CNNAE	Recon. MSE	.306 ± .002	.312 ± .003	.312 ± .005	.345 ± .012
		Imput. MSE	-	.292 ± .028	.319 ± .027	.328 ± .010
		Relative Imput. MSE	-	−.020 ± .028	.007 ± .027	−.017 ± .016
	M-CNNAE	Recon. MSE	.294 ± .006	.303 ± .006	.309 ± .007	.364 ± .012
		Imput. MSE	-	.283 ± .034	.314 ± .031	.351 ± .006
		Relative Imput. MSE	-	−.020 ± .035	.005 ± .032	−.014 ± .013
Pt 3	Baseline	Recon. MSE	1.05	1.09 ± .049	1.19 ± .054	1.24 ± .591
		Imput. MSE	-	.691 ± .393	.537 ± .212	.748 ± .319
		Relative Imput. MSE	-	−.397 ± .396	−.652 ± .219	−.488 ± .671
	CNNAE	Recon. MSE	.674 ± .001	.729 ± .001	.795 ± .001	.878 ± .379
		Imput. MSE	-	.212 ± .006	.233 ± .008	.542 ± .360
		Relative Imput. MSE	-	−.517 ± .006	−.562 ± .008	−.335 ± .523
	M-CNNAE	Recon. MSE	.663 ± .003	.719 ± .004	.786 ± .003	.869 ± .374
		Imput. MSE	-	.212 ± .007	.232 ± .009	.542 ± .355
		Relative Imput. MSE	-	−.507 ± .008	−.554 ± .009	−.327 ± .516
Pt 4	Baseline	Recon. MSE	.328	.329 ± .002	.332 ± .009	.333 ± .006
		Imput. MSE	-	.308 ± .026	.306 ± .036	.338 ± .006
		Relative Imput. MSE	-	−.021 ± .026	−.026 ± .037	.006 ± .008
	CNNAE	Recon. MSE	.250 ± .001	.254 ± .004	.254 ± .007	.255 ± .006
		Imput. MSE	-	.249 ± .016	.269 ± .030	.303 ± .004
		Relative Imput. MSE	-	−.005 ± .016	.015 ± .031	.048 ± .007
	M-CNNAE	Recon. MSE	.250 ± .003	.253 ± .005	.254 ± .008	.260 ± .003
		Imput. MSE	-	.244 ± .018	.269 ± .029	.301 ± .004
		Relative Imput. MSE	-	−.008 ± .019	.014 ± .030	.041 ± .005
Pt 5	Baseline	Recon. MSE	3.51	3.12 ± 1.02	2.95 ± 1.33	3.34 ± 1.55
		Imput. MSE	-	3.54 ± 4.20	3.39 ± 2.28	2.68 ± .520
		Relative Imput. MSE	-	.425 ± 4.32	.439 ± 2.63	−.662 ± 1.63
	CNNAE	Recon. MSE	1.92 ± .012	1.75 ± .467	1.74 ± .656	2.38 ± .987
		Imput. MSE	-	3.45 ± 4.18	2.72 ± 2.61	1.57 ± .966
		Relative Imput. MSE	-	.170 ± 4.20	.988 ± 2.69	−.811 ± 1.38
	M-CNNAE	Recon. MSE	1.89 ± .005	1.73 ± .465	1.72 ± .654	2.36 ± .990
		Imput. MSE	-	3.43 ± 4.16	2.69 ± 2.60	1.55 ± .964
		Relative Imput. MSE	-	1.70 ± 4.18	.969 ± 2.68	−.818 ± 1.38
Pt 6	Baseline	Recon. MSE	7.92	7.59 ± .598	6.95 ± 1.84	6.35 ± 1.72
		Imput. MSE	-	7.29 ± 3.06	7.00 ± 3.94	6.83 ± 1.10
		Relative Imput. MSE	-	−.345 ± 3.11	.046 ± 4.35	.479 ± 2.04
	CNNAE	Recon. MSE	6.13 ± .006	6.39 ± .300	6.17 ± 1.14	6.14 ± 1.88
		Imput. MSE	-	3.93 ± 2.70	6.10 ± 4.57	6.29 ± 1.87
		Relative Imput. MSE	-	−2.46 ± 2.71	−.077 ± 4.71	.151 ± 2.65
	M-CNNAE	Recon. MSE	6.10 ± .016	6.35 ± .299	6.14 ± 1.14	6.10 ± 1.88
		Imput. MSE	-	3.94 ± 2.71	6.08 ± 4.57	6.27 ± 1.88
		Relative Imput. MSE	-	−2.41 ± 2.72	−.059 ± 4.71	.166 ± 2.65

Table 10: **MSE of Deep Neural Imputation Methods (Pt. 1-6).** We show the time series mean squared error for the baseline, CNNAE, and M-CNNAE. The top row for each method represents reconstruction, the middle row imputation, and the bottom row imputation minus reconstruction. The mean is over 9 runs (3 runs of each model with 3 sets of missing data). Due to space, results for participants 7 to 12 are in Table 11. Lower MSE values are better.

			% Electrodes Missing			
			0%	10%	20%	50%
Pt 7	Baseline	Recon. MSE	.326	.331 ± .002	.327 ± .017	.373 ± .016
		Imput. MSE	-	.326 ± .035	.339 ± .017	.353 ± .014
		Relative Imput. MSE	-	−.005 ± .035	.012 ± .024	−.020 ± .021
	CNNAE	Recon. MSE	.264 ± .014	.262 ± .014	.263 ± .015	.296 ± .019
		Imput. MSE	-	.312 ± .047	.305 ± .021	.302 ± .019
		Relative Imput. MSE	-	.050 ± .049	.042 ± .026	.005 ± .027
	M-CNNAE	Recon. MSE	.240 ± .006	.242 ± .009	.245 ± .008	.282 ± .010
		Imput. MSE	-	.279 ± .048	.276 ± .014	.292 ± .011
		Relative Imput. MSE	-	.037 ± .049	.030 ± .016	.010 ± .015
Pt 8	Baseline	Recon. MSE	.386	.397 ± .001	.403 ± .009	.449 ± .005
		Imput. MSE	-	.386 ± .026	.411 ± .022	.441 ± .011
		Relative Imput. MSE	-	−.011 ± .026	.008 ± .024	−.008 ± .012
	CNNAE	Recon. MSE	.322 ± .001	.322 ± .002	.320 ± .005	.342 ± .021
		Imput. MSE	-	.366 ± .021	.372 ± .018	.392 ± .008
		Relative Imput. MSE	-	.044 ± .021	.052 ± .019	.050 ± .022
	M-CNNAE	Recon. MSE	.333 ± .001	.336 ± .002	.335 ± .006	.366 ± .014
		Imput. MSE	-	.376 ± .026	.384 ± .019	.398 ± .006
		Relative Imput. MSE	-	.040 ± .026	.049 ± .020	.032 ± .015
Pt 9	Baseline	Recon. MSE	.589	.577 ± .004	.546 ± .025	.523 ± .004
		Imput. MSE	-	.538 ± .060	.483 ± .060	.495 ± .044
		Relative Imput. MSE	-	−.039 ± .060	−.063 ± .065	−.028 ± .059
	CNNAE	Recon. MSE	.388 ± .001	.389 ± .005	.401 ± .006	.431 ± .021
		Imput. MSE	-	.401 ± .042	.364 ± .027	.403 ± .034
		Relative Imput. MSE	-	.012 ± .042	−.037 ± .028	−.028 ± .040
	M-CNNAE	Recon. MSE	.385 ± .004	.387 ± .006	.403 ± .008	.448 ± .017
		Imput. MSE	-	.396 ± .038	.366 ± .024	.427 ± .035
		Relative Imput. MSE	-	.008 ± .038	−.037 ± .025	−.021 ± .039
Pt 10	Baseline	Recon. MSE	.447	.451 ± .002	.407 ± .022	.392 ± .014
		Imput. MSE	-	.433 ± .007	.582 ± .033	.410 ± .015
		Relative Imput. MSE	-	−.018 ± .007	.069 ± .040	.018 ± .021
	CNNAE	Recon. MSE	.325 ± .003	.321 ± .003	.323 ± .005	.340 ± .010
		Imput. MSE	-	.372 ± .022	.364 ± .015	.341 ± .014
		Relative Imput. MSE	-	.051 ± .022	.025 ± .016	.001 ± .017
	M-CNNAE	Recon. MSE	.314 ± .004	.312 ± .004	.316 ± .005	.335 ± .013
		Imput. MSE	-	.362 ± .017	.393 ± .013	.338 ± .013
		Relative Imput. MSE	-	.050 ± .017	.026 ± .014	.003 ± .018
Pt 11	Baseline	Recon. MSE	.561	.554 ± .015	.550 ± .019	.572 ± .025
		Imput. MSE	-	.558 ± .024	.582 ± .030	.565 ± .022
		Relative Imput. MSE	-	.004 ± .028	.032 ± .036	−.007 ± .033
	CNNAE	Recon. MSE	.315 ± .002	.312 ± .003	.313 ± .004	.314 ± .013
		Imput. MSE	-	.385 ± .017	.364 ± .008	.394 ± .008
		Relative Imput. MSE	-	.073 ± .017	.051 ± .009	.080 ± .015
	M-CNNAE	Recon. MSE	.328 ± .007	.329 ± .008	.339 ± .010	.391 ± .014
		Imput. MSE	-	.408 ± .013	.393 ± .008	.457 ± .015
		Relative Imput. MSE	-	.079 ± .015	.053 ± .013	.065 ± .021
Pt 12	Baseline	Recon. MSE	.357	.359 ± .001	.366 ± .006	.374 ± .006
		Imput. MSE	-	.354 ± .014	.370 ± .007	.388 ± .008
		Relative Imput. MSE	-	−.005 ± .014	.004 ± .009	.014 ± .010
	CNNAE	Recon. MSE	.294 ± .001	.299 ± .002	.299 ± .003	.304 ± .003
		Imput. MSE	-	.272 ± .019	.299 ± .006	.319 ± .005
		Relative Imput. MSE	-	−.027 ± .019	.000 ± .007	.015 ± .006
	M-CNNAE	Recon. MSE	.299 ± .003	.306 ± .004	.306 ± .004	.318 ± .002
		Imput. MSE	-	.280 ± .020	.307 ± .004	.337 ± .007
		Relative Imput. MSE	-	−.026 ± .020	.001 ± .006	.019 ± .007

Table 11: **MSE of Deep Neural Imputation Methods (Pt. 7-12).** We show the time series mean squared error for the baseline, CNNAE, and M-CNNAE. The top row for each method represents reconstruction, the middle row imputation, and the bottom row imputation minus reconstruction. The mean is over 9 runs (3 runs of each model with 3 sets of missing data). Due to space, results for participants 1 to 6 are in Table 10. Lower MSE values are better.