DeCorrNet: Enhancing Neural Decoding Performance by Eliminating Correlations in Noise

Xianhan Tan[1,2,3], **Yu Qi**[2,3,4,1*], **Yueming Wang**[1]

[1]The College of Computer Science and Technology, Zhejiang University, China
[2]MOE Frontier Science Center for Brain Science and Brain-machine Integration, Zhejiang University, China
[3]Affiliated Mental Health Center & Hangzhou Seventh People's Hospital, Zhejiang University, China
[4]State Key Lab of Brain-Machine Intelligence, Zhejiang University, China
{txxxl, qiyu, ymingwang}@zju.edu.cn

Abstract

Neural decoding, which transforms neural signals into motor commands, plays a key role in brain-computer interfaces (BCIs). Existing neural decoding approaches mainly rely on the assumption of independent noises, which could perform poorly in case the assumption is invalid. However, correlations in noises have been commonly observed in neural signals. Specifically, noise in different neural channels can be similar or highly related, which could degrade the performance of those neural decoders. To tackle this problem, we propose the DeCorrNet, which explicitly removes noise correlation in neural decoding. DeCorrNet could incorporate diverse neural decoders as an ensemble module to enhance the neural decoding performance. Experiments with benchmark BCI datasets demonstrated the superiority of DeCorrNet and achieved state-of-the-art results.

Introduction

Recent advances in brain-computer interfaces (BCIs), especially neural decoding (Liu et al. 2023b; Card et al. 2023; Wilson et al. 2020; Moses et al. 2021; Qi et al. 2019, 2022), have demonstrated great potential in restoring motor or speech abilities for individuals with paralysis (Willett et al. 2023; Metzger et al. 2023; Willett et al. 2021). Despite this progress, achieving stable and high-performance neural decoding in BCI systems remains a formidable challenge.

An important issue in neural decoding is dealing with noise in neural signals. Most of the noise in neural signals is independent and identically distributed (*i.i.d*). Specifically, cortical neurons encode information in certain spike patterns. However, given the non-stationary nature of the neural system, there are variations in the spike patterns even with repeated trials, which can be considered as *i.i.d* noise. Typically, *i.i.d* noise can be suppressed by averaging the recordings from a large number of neurons. However, there is still a kind of correlated noise in neural signals, that is, there are correlation patterns across neurons. In this case, the correlation patterns cannot be removed by averaging which leaves variability in the signal. Therefore, the amount of information in the neural population may be limited due to the presence of correlated noise, significantly degrading the performance of neural decoding. This noise is overlooked and difficult to remove by traditional denoising methods (Moreno-Bote et al. 2014; Lai et al. 2023).

Existing studies have shown that functional interactions between neurons result in rich correlations in neuronal activities (Uddin 2020; Stringer et al. 2021; Kanitscheider, Coen-Cagli, and Pouget 2015; Panzeri et al. 2022; Valente et al. 2021; Kafashan et al. 2021; Ito and Murray 2022). From the perspective of neural decoding, these correlations can be divided into two parts (Figure 1a). The basic part, named *signal correlations*, reflects the true connectivity and interaction patterns between different neurons in response to different tasks; The noise part, named *noise correlations*, refers to the correlations of fluctuating in neuronal responses under the same task, which cause information limits in specific situations. Figure 1b illustrates the impact of noise correlations on information encoding. When the intra-task and inter-task neuronal responses covary in the same direction, there will be overlap between different task responses, which leads to task confusion (Figure 1b). As is shown in Figure 1c, we further visualize the correlation matrix of responses from five neurons across three tasks. Given the noise correlations in two clusters of (N1, N2) and (N3, N4, N5), neural signals from three different tasks reveal similar dominating correlation patterns. When noise correlation patterns appear, information from signal correlations could be masked by noise correlations. Given the full dataset and the labels of trials, the *shuffling* method (Panzeri et al. 2022) can be applied to eliminate the effect of noise correlation. Specifically, we shuffle the trial indices independently for each neuron across trials of the same labels. As a result, correlations that are unrelated to the tasks can be suppressed. We use the Wasserstein distance (Panaretos and Zemel 2019) to measure the distance between the correlation matrices across three tasks and each task. After shuffling, the distance is significantly increased and the correlation clusters are mostly eliminated, thereby signal correlations can be enhanced and highly improved task classification performance. Although *shuffling* method can solve the problem, it needs all the information of the dataset, which is difficult to use in practice.

Mostly denoising and decoding models (Romero, Piñol, and Vázquez-Seisdedos 2021; Ho, Jain, and Abbeel 2020; Zhang and Zhang 2019) assume that noise is independent,

*Corresponding author

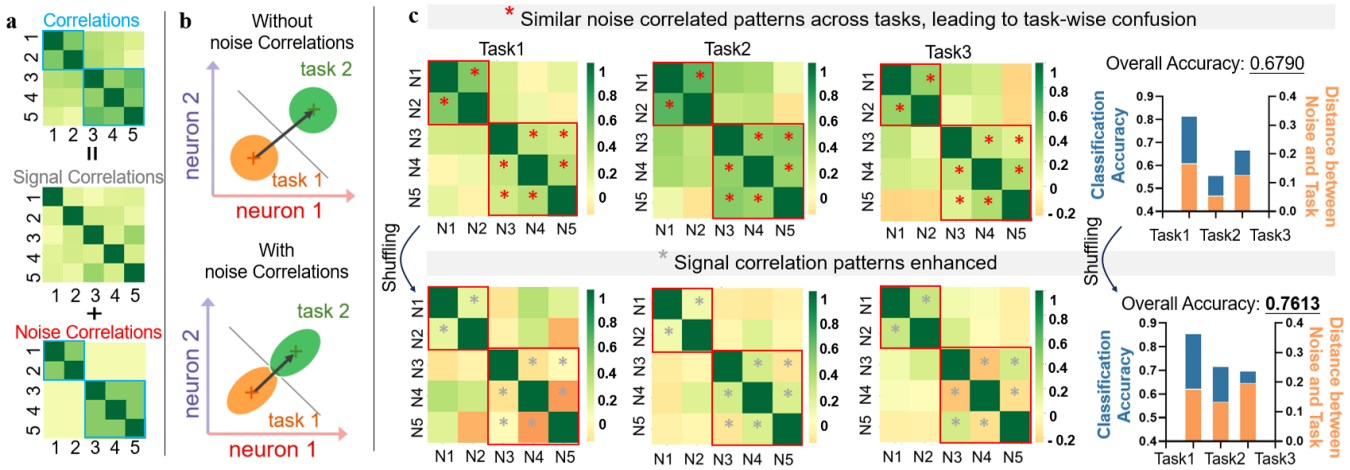

Figure 1: Analysis of neuronal correlations. (a) Neuronal correlations contain both signal correlations and noise correlations. (b) A conceptual diagram to illustrate how noise correlation affects decoding performance. We plot the responses of the two neurons as variables respectively in the coordinate axis, while plotting the covariance ellipse of the two variables. Black arrows show the direction of signal correlation. (c) Left: We plot the correlation matrix for neurons N1-N5 across 3 tasks, highlighting similar patterns with red asterisks. By the shuffling method, these noise correlations are removed (indicated by gray asterisks). Right: Task classification performance and distance between signal correlation matrix (all tasks) and noise correlation matrix (single task) before or after shuffling.

often overlooking correlations in noise, which can lead to a decline in decoding performance. To address this issue, we propose DeCorrNet, a denoising framework that explicitly eliminates the correlation noise in an end-to-end manner. To eliminate noise correlations in neural signals, our basic idea is to separate the two kind of correlations by learning the information of different tasks and the interactions between neurons. As mentioned above, noise correlations limit information only if it is in the same direction as signal correlation. Considering that the exact calculation of the changing direction is difficult under finite neural records, we uniformly minimize the noise correlation of all directions. Concretely, DeCorrNet preserves basic information by maximizing the signal correlations of the input. At the same time, all of the noise correlations are minimized to polish the correlation pattern. These two parts are weighted by learnable dynamic parameters and optimized collaboratively. The tasks of input is subsampled in each iteration, which fully learns the correlation representation between different tasks and expands the data size. The proposed decorrelation loss and the DeCorrNet framework can be used as a plug-and-play module with various decoding models to enhance the decoding performance. We evaluate our proposed model on a simulated dataset and five neural datasets with diverse tasks: a vision animals dataset, a human handwriting dataset, a human speech dataset, and two monkey Center-Out datasets. Experiments demonstrate that DeCorrNet achieves stable and significant performance improvement compared with other denoising methods. Furthermore, as a plugin, DeCorrNet improves performance on all tasks and all decoding models.

Related Work

Neural Decoding The goal of neural decoding is to generate predictions for an input of neural signals. Predictions include continuous coordinates, velocities as well as discrete action and articulation categories. This depends on the design and goals of the experimental paradigm. Due to the limitation of the amount of data, early studies tended to use non-neural network methods (SVM (Livezey, Bouchard, and Chang 2019), NaiveBayes (Sereshkeh et al. 2017)) for decoding with good results. To further improve the decoding performance, more data has been collected and decoded using neural network methods such as MLP (Wang et al. 2017), GRU (Wilson et al. 2020), GNN (Chen et al. 2024c,d,b) and Transformer (Chen et al. 2023). Although all these efforts have achieved good or even very good decoding performance, the performance is highly dependent on the data quality. In other words, the signal-to-noise ratio of the data determines the decoding performance on the line.

Neural Signal Denoising Denoising is a key preprocessing step for neural signals before decoding. Extensive research (Duan et al. 2023; Romero, Piñol, and Vázquez-Seisdedos 2021; Jiang et al. 2023; Zhang and Zhang 2019) has been conducted to eliminate noise from neural signals. Although these efforts have yielded good results, these designs for noise are too empirical. For example, DS-DDPM(Duan et al. 2023) proposes to utilize a diffusion process with multiple steps for denoising. However, diffusion models mainly focus on Gaussian noise and therefore cannot effectively reduce correlated noise from neural signals. Deepfilter(Romero, Piñol, and Vázquez-Seisdedos 2021) designs a deep learning network to remove noise and FECAM(Jiang et al. 2023) proposes a frequency attention

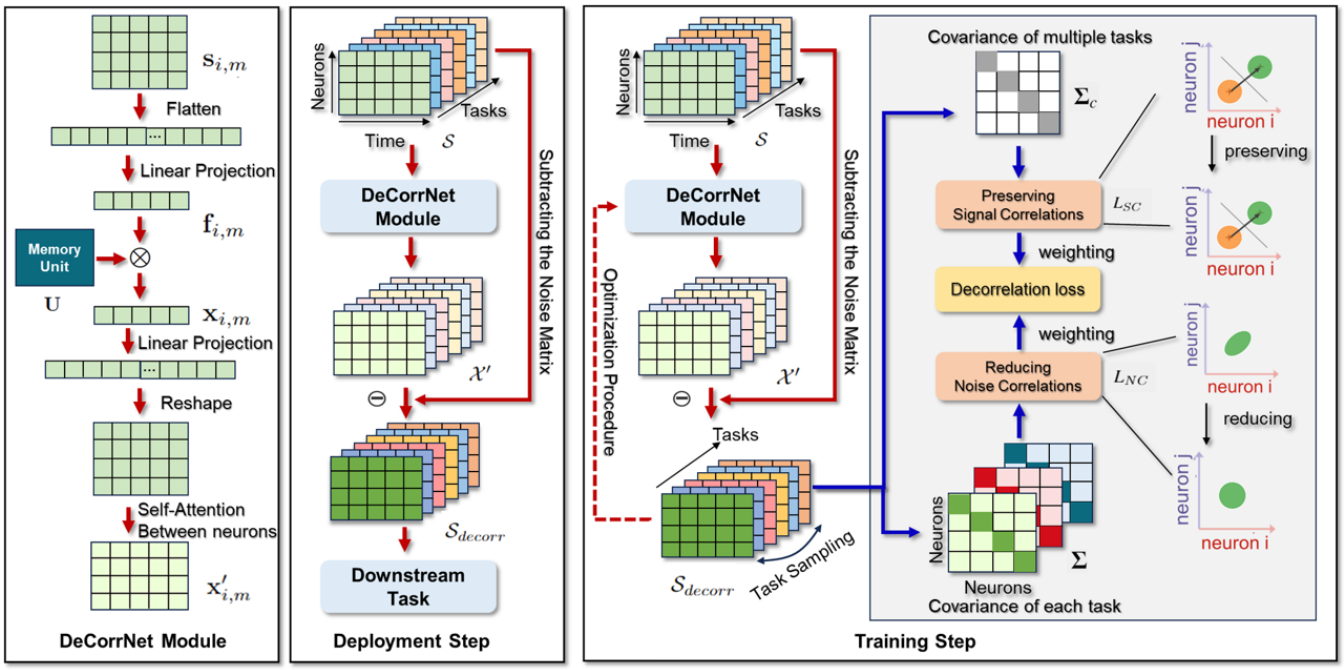

Figure 2: The framework of DeCorrNet. Left: we plot the network structure of the DeCorrNet Module. Middle: we plot the deployment process of DeCorrNet, which needs a well-trained DeCorrNet Module. Right: we plot the training process of DeCorrNet. The gray rectangle contains the Decorrelation Loss calculation during training.

Notation	Description
$\mathcal{S}$	The training dataset of M tasks
$\mathbf{s}_{i,m}$	The i th data point in task m
$\mathcal{F}$	The feature matrix after linearly projection
$\mathbf{U}$	The external memory unit
$\mathcal{X}$	The feature matrix after external attention
$\mathcal{X}'$	The predicted noise
$\mathbf{\Sigma}_{\text{off-diag}}$	The off-diagonal elements
$\mathbf{\Sigma}_m$	The correlation matrix for task m
$\mathbf{\Sigma}_c$	The correlation matrix for all tasks

Table 1: Notation Table

mechanism to eliminate high-frequency noise. These methods all rely on the assumption of independent noise, and once correlations appear in the noise, these methods cannot effectively distinguish the noise from the signal. These methods do not design the denoising process for the correlations of the neural response, which limits the performance of neural decoding and leaves a huge gap.

The Framework of DeCorrNet

Our method explicitly calculates the neuronal correlations within signals for each task and across multiple tasks. By assuming the noise correlation is stable and predictable, DeCorrNet refines the signal by continuously subtracting prediction noise, ensuring that noise correlations are minimized while preserving task-dependent correlations.

Our DeCorrNet consists of the DeCorrNet module, which is divided into two steps: training and deployment. The DeCorrNet module can take neural signals as input and output the corresponding noise with correlations. Firstly, by taking neural signals of different tasks as input, the information of different tasks is encoded and stored. Then the interactions between neurons are learned through the self-attention module. Finally, the predicted noise is obtained. In the training step, we subtract the predicted noise from the original signal and optimize this denoised signal by our decorrelation loss. In the decorrelation loss, we keep task-related correlations while eliminating noise correlations. In the deployment step, we denoise the signals with the trained DeCorrNet and use the denoised signals for different downstream tasks. The pseudo-code is presented in Algorithm 1. The notations are presented in Table 1.

DeCorrNet Module

Encoding Task Information in Neural Responses Consider a training dataset with M tasks, and each task have N data points: $\mathcal{S} = [\{\mathbf{s}_{i,1}\}_{i=1}^{N}, \{\mathbf{s}_{i,2}\}_{i=1}^{N}, \ldots, \{\mathbf{s}_{i,M}\}_{i=1}^{N}]$, where $\mathbf{s}_{i,m} \in \mathbb{R}^{Ne \times T}$, Ne is the number of neurons and T represents the time bins of neural activity. For each task m, we begin by flattening $\mathbf{s}_{i,m}$ to form $\mathbf{s}'_{i,m} \in \mathbb{R}^{d}$, where $d = Ne \times T$. The task encoder then linearly projects this input into a feature matrix $\mathcal{F} = \{\mathbf{f}_{i,m}\}_{i=1}^{N}$, where $\mathbf{f}_{i,m} \in \mathbb{R}^{d'}$ and d' is the feature dimension.

To store information regarding different tasks, we encode task information by interacting the feature matrix with an external memory unit $\mathbf{U} \in \mathbb{R}^{d' \times d'}$, which is analogous to

external attention mechanisms (Guo et al. 2022; Liang et al. 2023, 2024).

$$\mathcal{X} = \{Norm(\mathbf{f}_{i,m}\mathbf{U})\}_{i=1}^{N} = \{\mathbf{x}_{i,m}\}_{i=1}^{N} \tag{1}$$

Here, the memory unit $\mathbf{U}$ is a linear layer without bias, preserving the feature dimension while enhancing the encoding with task-specific information. The normalization (denoted as *Norm*) ensures that the output features maintain numerical stability and are appropriately scaled for subsequent processing.

Finally, the task encoder linearly projects the output feature matrix back to dimension d and reshapes it to the original dimensions, thereby ensuring that the encoded task information remains consistent with the original input format.

Learning the Interactions Between Neurons Following the encoding of task information via the task encoder, a self-attention block (Vaswani et al. 2017) is applied to capture the interactions between neurons. Consider a single data point i in task m: $\mathbf{x}_{i,m} = [\mathbf{x}_{i,m}^1, \mathbf{x}_{i,m}^2, \ldots, \mathbf{x}_{i,m}^{Ne}]$ represent the features of n neurons extracted by the task encoder. We first linearly project these features into a query matrix $\mathbf{Q}_{i,m} = [\mathbf{q}_{i,m}^1, \mathbf{q}_{i,m}^2, \ldots, \mathbf{q}_{i,m}^{Ne}]$, a key matrix $\mathbf{K}_{i,m} = [\mathbf{k}_{i,m}^1, \mathbf{k}_{i,m}^2, \ldots, \mathbf{k}_{i,m}^{Ne}]$, and a value matrix $\mathbf{V}_{i,m} = [\mathbf{v}_{i,m}^1, \mathbf{v}_{i,m}^2, \ldots, \mathbf{v}_{i,m}^{Ne}]$. The self-attention operation is then expressed as:

$$\mathbf{x}'_{i,m} = \textit{Softmax}\left(\frac{\mathbf{Q}_{i,m}\mathbf{K}_{i,m}^T}{\sqrt{d_k}}\right)\mathbf{V}_{i,m} \tag{2}$$

Combining all data points and tasks, we get the output $\mathcal{X}'$.

The Decorrelation Loss

To address noise correlations while preserving signal correlations, we introduce a decorrelation loss function. This loss function comprises two components: a noise correlation term, which is minimized, and a signal correlation term, which is maximized.

The final output $\mathcal{X}'$ maintains consistency with the input feature's dimensions. The decorrelated neural signal is obtained by subtracting the final output from the original neural signal:

$$\mathcal{S}_{decorr} = \mathcal{S} - \textit{Norm}(\mathcal{X}') \tag{3}$$

Noise Correlation Term. The noise correlation can be computed using the decorrelated signals $\mathcal{S}_{decorr,m} = \{\mathbf{s}_{i,m}^{de}\}_{i=1}^{N}$ for each task m, where $\mathbf{s}_{i,m}^{de} \in \mathbb{R}^{Ne \times T}$. Firstly, the N data points of $\mathcal{S}_{decorr,m}$ can be concatenated as $\mathcal{S}'_{decorr,m} \in \mathbb{R}^{Ne \times Total}$, where $Total = N \times T$. The covariance matrix of neurons $\mathbf{\Sigma}_m$ is then calculated by $\mathcal{S}'_{decorr,m}$.

To capture the correlations among all neurons, we use the Frobenius norm of the off-diagonal elements of the covariance matrix:

$$L_{NC} = \frac{1}{2}\left\|\mathbf{\Sigma}_{\text{off-diag},m}\right\|_F^2 \tag{4}$$

Algorithm 1: The training and deployment procedures.

Require: $\mathcal{S} = [\{\mathbf{s}_{i,1}\}_{i=1}^{N_1}, \{\mathbf{s}_{i,2}\}_{i=1}^{N_1}, \ldots, \{\mathbf{s}_{i,M}\}_{i=1}^{N_1}]$: training data, $\mathcal{S}_* = [\{\mathbf{s}_{i,1}\}_{i=1}^{N_2}, \{\mathbf{s}_{i,2}\}_{i=1}^{N_2}, \ldots, \{\mathbf{s}_{i,M}\}_{i=1}^{N_2}]$: data to be denoised. $N_1 : N_2 = 2 : 8$.

Ensure: $\mathcal{S}_{decorr}$: the denoised data, which can be used for downstream tasks and achieve better performance.

Training Phase:

for each optimizer step do
  uniformly and randomly sample $m_0 \in [5, M]$ and obtain data $\mathcal{S}_{m_0} \subseteq \mathcal{S}$
  calculate the noise $\mathcal{X}'$ and obtain the denoised data $\mathcal{S}_{decorr,m_0} = \mathcal{S}_{m_0} - Norm(\mathcal{X}')$
  calculate neuron covariance matrix $\mathbf{\Sigma}_1, \mathbf{\Sigma}_2, ..., \mathbf{\Sigma}_{m_0}$ for each task, and $\mathbf{\Sigma}_c$ for all m_0 tasks concatenated
  optimize objective function in Eq. (8)

Deployment Phase:

calculate the noise of $\mathcal{S}_*$ and obtain the denoised data $\mathcal{S}_{decorr} = \mathcal{S}_* - Norm(\mathcal{X}_*{}')$
train and test decoding models in downstream tasks using the denoised data $\mathcal{S}_{decorr}$

This term penalizes the presence of off-diagonal elements in the covariance matrix, which correspond to noise correlations between different neurons.

Signal Correlation Term. Consider the complete $\mathcal{S}_{decorr} = [\{\mathbf{s}_{i,1}^{de}\}_{i=1}^{N}, \{\mathbf{s}_{i,2}^{de}\}_{i=1}^{N}, \ldots, \{\mathbf{s}_{i,M}^{de}\}_{i=1}^{N}]$, Similarly, each task of $\mathcal{S}_{decorr}$ can be concatenated as: $\mathcal{S}'_{decorr} = [\mathbf{s}_{i,1}^{de'}, \mathbf{s}_{i,2}^{de'}, \ldots, \mathbf{s}_{i,M}^{de'}]$, where $\mathbf{s}_{i,m}^{de'} \in \mathbb{R}^{Ne \times Total}$. And the M tasks of $\mathcal{S}'_{decorr}$ can be concatenated again as $\mathcal{S}''_{decorr} \in \mathbb{R}^{Ne \times MTotal}$, where $MTotal = M \times Total$, Then the covariance matrix of $\mathcal{S}''_{decorr}$ is calculated as $\mathbf{\Sigma}_c$.

Finally, the signal correlation term can be obtained as mentioned above:

$$L_{SC} = \frac{1}{2}\left\|\mathbf{\Sigma}_{c,\ \textit{off-diag}}\right\|_F^2 \tag{5}$$

This term ensures that the model retains and even enhances the meaningful correlations that are indicative of the underlying signal structure across different tasks.

Task Sampling. By task sampling, the DeCorrNet can learn rich inter-task correlations while augmenting the training samples. Firstly, we subsample tasks multiple times to augment the data and fully learn correlation patterns between different tasks. Consider M tasks, we uniformly and randomly sample $m_0 \in [5, M]$ tasks per iteration. The noise correlation term for m_0 tasks is computed separately as $NC_1, NC_2, \ldots, NC_{m_0}$, and summed as the noise correlation loss term. The signal correlation term is computed by concatenating m_0 decorrelated signals, following the aforementioned procedure.

Overall Loss Function. To balance the multiple optimization objectives, we introduce trainable parameters

$[p_1, p_2, \ldots, p_{m_0+1}]$ such that $\sum_{i=1}^{t_0+1} p_i = 1$.

The noise correlation loss term is modified to a weighted sum of NCs:

$$L_{NC} = \sum_{i=1}^{m_0} p_i \left\| \mathbf{\Sigma}_{\text{off-diag},i} \right\|_F^2 \tag{6}$$

Similarly, the signal correlation loss term is weighted by the last parameter p_{m_0+1}:

$$L_{SC} = p_{m_0+1} \left\| \mathbf{\Sigma}_{c,\ \textit{off-diag}} \right\|_F^2 \tag{7}$$

Combining these terms, the overall loss function is defined as:

$$L = \sum_{i=1}^{m_0} p_i \left\| \mathbf{\Sigma}_{\text{off-diag},i} \right\|_F^2 - p_{t_0+1} \left\| \mathbf{\Sigma}_{c,\ \textit{off-diag}} \right\|_F^2 \tag{8}$$

Figure 2 illustrates the DeCorrNet training and deployment processes conceptually, and Algorithm 1 specifies them in detail. In the training phase, the first step is to randomly sample m_0 tasks from the M tasks and divide the data batch according to m_0 tasks. DeCorrNet computes task-related information as well as neuron-related information for each input and predicts noise in the data. Subtracting the predicted noise, we get the denoised data. Subsequently, we compute the Frobenius norm for the off-diagonal elements of the neural response correlation matrix for the denoised data. It includes the neuron correlation matrix within each task and for all tasks concatenated together. During optimization, the former is minimized while the latter is maximized. The two parts are weighted by learnable parameters. In the deployment phase, we can obtain the denoised data by well-trained DeCorrNet, which can be used to train decoding models in downstream tasks.

Experiments and Results

Benchmark Datasets and Methods

For experiments, we evaluate the DeCorrNet on five real neural datasets and a simulated neural dataset with diverse classification tasks. We compare the DeCorrNet with four denoising models. After denoising, we use six decoding models to show the effectiveness of our model.

Datasets We used multiple datasets, including simulated dataset, vision dataset with animals (Kafashan et al. 2021), handwriting dataset (Willett et al. 2021), speech dataset (Willett et al. 2023) and center-out datasets: (M) (Ma et al. 2023) and (RS) (Sundiang, Hatsopoulos, and MacLean 2023). The simulated signals were generated by a Linear-nonlinear-Poisson (LNP) population model (Corrado et al. 2005), with neural signals of 1000 neurons corresponding to two different stimulus orientations, and noise correlations were added to the signals. For the other datasets, we preprocessed the data in the same way as in references.

Data	Denoisng	SVM	GRU	MLP	MTCN	TimesNet
Handwriting I	Raw	92.8	91.6	90.5	84.6	89.6
	Diffusion	92.8	90.3	90.0	85.6	83.7
	WT	91.2	93.1	81.3	81.9	87.6
	DeepFilter	87.5	82.1	83.5	83.5	86.3
	FECAM	88.3	85.6	83.5	87.7	86.1
	MECG-E	92.3	93.1	89.6	86.2	88.3
	TCDAE	89.1	92.3	87.5	83.2	85.5
	Ours	**95.8**	**94.4**	**92.8**	**86.9**	**90.8**
Speech	Raw	61.1	58.8	57.4	60.8	63.3
	Diffusion	58.3	59.0	55.6	55.3	61.3
	WT	58.9	58.8	53.1	60.3	58.3
	DeepFilter	58.1	54.4	61.2	57.5	64.8
	FECAM	54.7	57.9	52.0	58.1	56.1
	MECG-E	69.1	67.4	66.5	72.6	73.0
	TCDAE	65.8	68.2	66.1	69.5	71.3
	Ours	**70.3**	**69.5**	**68.3**	**75.1**	**77.6**
Center-Out (M)	Raw	73.1	85.7	84.6	82.1	82.4
	Diffusion	74.8	83.4	83.4	81.0	79.7
	WT	75.4	84.0	81.5	82.2	80.0
	DeepFilter	68.9	75.2	78.0	73.2	79.3
	FECAM	75.6	80.8	79.0	79.8	78.4
	Ours	**85.5**	**90.8**	**95.1**	**93.3**	**95.3**
Center-Out (RS)	Raw	95.8	92.6	93.4	89.7	98.6
	Diffusion	94.2	92.8	91.5	88.4	97.6
	WT	95.3	98.3	96.4	98.1	98.3
	DeepFilter	93.4	92.8	96.3	87.6	88.5
	FECAM	97.2	95.4	94.8	95.6	97.6
	Ours	**98.3**	**98.7**	**98.5**	**98.5**	**98.7**

Table 2: Comparsion with different denoisng methods. We computed classification accuracy (%) as metric.

Baseline Methods We compared multiple denoising models, including Wavelet Transform (WT) (Zhang and Zhang 2019), Diffusion (Ho, Jain, and Abbeel 2020), FECAM (Jiang et al. 2023), DeepFilter (Romero, Piñol, and Vázquez-Seisdedos 2021), TCDAE (Chen et al. 2024a) and MECG-E (Hung et al. 2024). To facilitate the comparison of the performance, we employ the parameter settings from the original references on each baseline method.

For decoding in downstream tasks, we used multiple decoding models to test the decoding performance before and after denoising, including SVM, GRU, MLP, ModernTCN (MTCN) (Luo and Wang 2024), TimesNet (Wu et al. 2022) and iTransformer (Liu et al. 2023a).

Experimental Setup Our model is implemented using Keras (Ketkar and Ketkar 2017) and trained on a single NVIDIA A100 40GB GPU. We employ the Adam optimizer (Kingma and Ba 2014) with a learning rate of 0.01. During the evaluation phase, we assess the decoding performance on two types of data: raw and decorrelated. This dual evaluation provides insights into the effectiveness of our decorrelation process. Given the limited number of samples in clinical human datasets, we use only 20% data for training, ensuring that our model is robust and generalizable even with small

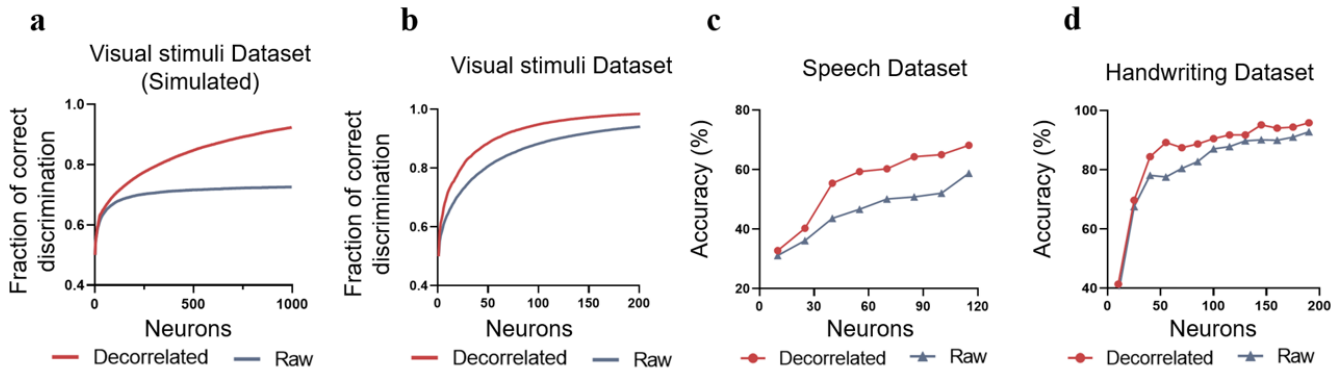

Figure 3: Decoding performance improvement on different datasets. (a) and (b) show the better virtual discrimination performance achieved by DeCorrNet, while (c) and (d) show better classification accuracies.

Dataset \ Model	SVM		GRU		MLP	
	R	D	R	D	R	D
Handwriting I	92.8	**95.8**	91.6	**94.4**	90.5	**92.8**
Handwriting II	66.0	**81.5**	67.5	**75.2**	55.1	**84.2**
Speech	61.1	**70.3**	58.8	**69.5**	57.4	**68.3**
Center-Out (M) I	69.2	**85.1**	67.7	**75.6**	72.5	**78.8**
Center-Out (M) II	65.0	**77.0**	67.0	**74.8**	76.2	**76.3**
Center-Out (M) III	69.7	**86.7**	82.0	**87.9**	79.2	**93.9**
Center-Out (M) IV	73.0	**90.9**	86.0	**91.1**	85.0	**96.2**
Center-Out (M) V	73.1	**85.5**	85.7	**90.8**	84.6	**95.1**
Center-Out (RS)	95.5	**98.3**	92.6	**98.7**	93.4	**98.5**

Dataset \ Model	MTCN		TimesNet		iTransformer	
	R	D	R	D	R	D
Handwriting I	84.6	**86.9**	89.6	**90.1**	93.4	**96.1**
Handwriting II	64.0	**84.1**	63.3	**88.2**	68.3	**86.0**
Speech	60.8	**75.1**	63.3	**77.6**	60.4	**71.6**
Center-Out (M) I	79.4	**82.7**	75.1	**81.5**	76.9	**88.9**
Center-Out (M) II	74.4	**82.0**	72.2	**78.1**	78.5	**89.6**
Center-Out (M) III	79.3	**91.8**	78.7	**96.5**	82.3	**96.6**
Center-Out (M) IV	85.0	**96.0**	84.3	**97.7**	84.0	**97.5**
Center-Out (M) V	82.1	**93.3**	82.4	**95.3**	83.1	**92.8**
Center-Out (RS)	97.8	**98.5**	98.6	**98.7**	95.8	**99.0**

Table 3: Classification accuracy (%) of DecorrNet incorporated with different neural decoders on all datasets. R denotes the raw data and D denotes the decorrelated data.

sample sizes.

For the simulated and vision datasets, we use the discrimination score (Kafashan et al. 2021) as the evaluation metric. For speech, handwriting, and Center-Out datasets, we use classification accuracy as the evaluation metric, and the accuracy is calculated by 10-fold cross-validation.

Comparison with Baseline

We compare our method with the candidate denoising methods on speech, handwriting, Center-Out (M), and Center-Out (RS) data. The results are shown in Table 2. Here, for Center-Out (M) and Handwriting, we use one day of data (Center-Out (M) V and Handwriting I) to evaluate performance. The performance improvement of DeCorrNet is more obvious. Other denoising methods achieve only a weak improvement, while some methods achieves even worse performance (DeepFilter in Center-out (M) dataset and WT in Speech dataset). The results suggest that these methods is according to an independent noise which is inconsistent with the noises with correlation.

Table 3 reports DeCorrNet's improvements in decoding performance with different decoding methods. Compared to the raw data, decorrelated data achieves remarkable decoding performance improvement on all decoding methods. For example, decorrelated data on the handwriting dataset achieves a classification accuracy of 96.1% with the iTransformer method, which outperforms the raw data by 2.7% . With the SVM method, the accuracy of decorrelated data is 95.8%, which still outperforms the raw data by 3.0%. For the speech dataset, decorrelated data achieves a greater performance improvement than raw data. For example, compared to raw data, decorrelated data has a 10.7% improvement with the GRU method, along with a 11.2% improvement with the iTransformer method. As is shown in Figure 3, on the simulated neural datasets and real neural datasets, the decoding performance of decorrelated data consistently outperforms the raw data. The results suggest that DeCorrNet can achieve performance improvement with different decoding methods and on different datasets.

Ablation Study

DeCorrNet Module. We conduct ablation study experiments on a handwriting dataset and decoding by iTransformer to investigate the effectiveness of two key design components of the DeCorrNet module, namely Task Encoder, and Self-Attention. We replace the Task Encoder with a simple fully connected network. The input and output dimensions are consistent with the Task Encoder. For Self-Attention, we remove this part directly. The results are summarized in Table 4, where we can see that removing both the Task Encoder and Self-Attention significantly hurt the per-

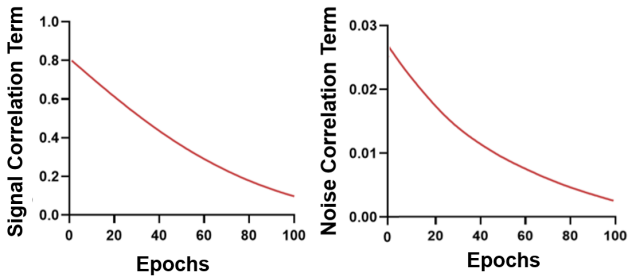

Figure 4: The weights of signal loss term and noise loss term change with the training process.

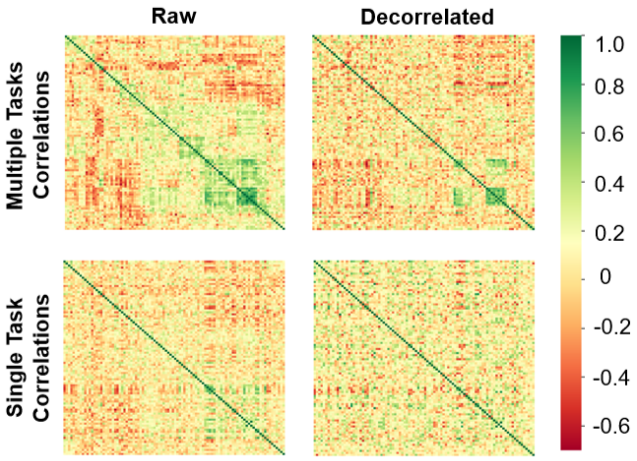

Figure 5: The correlation matrix across multiple tasks before and after decorrelation (the first row) and the correlation matrix for the single task before and after decorrelation (the second row).

formance. It shows that all of these are critical for learning task information and interaction between neurons.

Loss Weight. Also on the handwriting dataset, we investigate the importance of the loss of weight. We replace the weights with fixed weights: the weight of the signal correlation loss term is set to 0.5 and all of the noise correlation loss term is set to 0.1. In Table 4, the fixed weight cannot effectively weigh the magnitude of task correlation and noise correlation, and it is easy to be dominated by one of them. However, the loss weight in DeCorrNet (Figure 6) learns the equilibrium point and reduces the noise correlations while maintaining the signal correlations.

Analysis of Neuronal Correlation after Denoising

The visualization results show that the correlation of the neural response changes as expected by our method. In Figure 5, we visualize the change in the neuron correlation matrix before and after decorrelation. In the first row of Figure 5, we can see that signal correlation is still preserved after decorrelation. And the overall neuronal correlation coefficient remains almost unchanged (Figure 6, left). On the contrary, the correlation pattern in the correlation matrix for the

Component			ACC ↑	NC ↓
Task Encoder	Self-Attention	Loss weight		
✓	✓	✓	**96.1%**	**0.06**
	✓	✓	93.6%	0.07
✓		✓	94.5%	0.07
✓	✓		93.1%	0.08

Table 4: Ablation Study. The classification accuracy (ACC) and average of neuronal correlation coefficient (NC) are measured on the handwriting dataset.

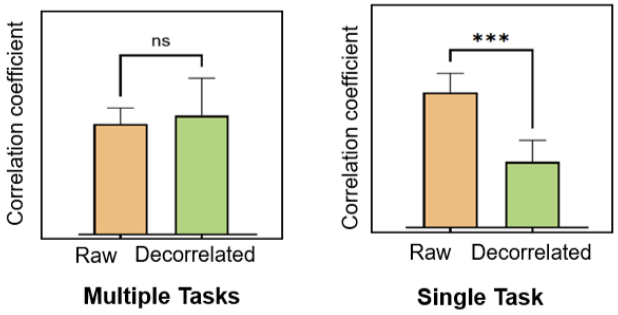

Figure 6: The average neuron response correlation coefficient across multiple tasks and single task.

single task disappears after decorrelation (the second row of Figure 5), while the noise correlation decreases significantly (Figure 6, right).

Conclusion and Limitation

We propose a simple, yet effective neural decorrelation network, called DeCorrNet, to explore the impact of correlations in noise on neural decoding. The major challenge here is how to eliminate correlations that are seen as noise without affecting useful information, given that the correlations between neurons are rich and complex. The proposed decorrelation (denoising) learning process with specially designed explicitly constrains the correlation of neurons under a single task after denoising, and retains the correlation between different tasks, so as to polish a high task-related and information-enhanced correlation pattern. Our findings claimed that using only a small portion of neural signals for denoising training can effectively denoise the remaining data and achieve better decoding performance. Therefore, we think our DeCorrNet provides potential evidence for the limitation of noise correlations on decoding performance, and propose that correlation-based learning can separate different types of correlation from neural signals.

However, our work only focus on eliminating the noise correlation between neurons. In fact, the noise correlation can also appear in signals of different time periods and we have not yet solved it. To effectively remove all the correlation noise, deriving the formula solution of correlation noise is another interesting direction for future work. Overall, we hope the proposed neural decorrelation framework will inspire new intelligence paradigms and provide a tool for neuroscience.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62276228), the Key Research and Development Program of Zhejiang (2023C03001), the Zhejiang Provincial Natural Science Foundation of China (LR24F020002), and the Fundamental Research Funds for the Central Universities.

References

Card, N. S.; Wairagkar, M.; Iaccobacci, C.; Hou, X.; Singer-Clark, T.; Willett, F. R.; Kunz, E. M.; Fan, C.; Vahdati Nia, M.; Deo, D. R.; et al. 2023. An accurate and rapidly calibrating speech neuroprosthesis. *medRxiv*, 2023–12.

Chen, J.; Zhang, Y.; Pan, Y.; Xu, P.; and Guan, C. 2023. A transformer-based deep neural network model for SSVEP classification. *Neural Networks*, 164: 521–534.

Chen, M.; Li, Y.; Zhang, L.; Liu, L.; Han, B.; Shi, W.; and Wei, S. 2024a. Elimination of Random Mixed Noise in ECG using Convolutional Denoising Autoencoder with Transformer Encoder. *IEEE Journal of Biomedical and Health Informatics*.

Chen, W.; Liu, Y.; Zhang, Z.; Zhuang, F.; and Zhong, J. 2024b. Modeling Adaptive Inter-Task Feature Interactions via Sentiment-Aware Contrastive Learning for Joint Aspect-Sentiment Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17781–17789.

Chen, W.; Wu, Y.; Zhang, Z.; Zhuang, F.; He, Z.; Xie, R.; and Xia, F. 2024c. FairGap: Fairness-aware Recommendation via Generating Counterfactual Graph. *ACM Transactions on Information Systems*, 42(4): 1–25.

Chen, W.; Yuan, M.; Zhang, Z.; Xie, R.; Zhuang, F.; Wang, D.; and Liu, R. 2024d. FairDgcl: Fairness-aware Recommendation with Dynamic Graph Contrastive Learning. *arXiv preprint arXiv:2410.17555*.

Corrado, G. S.; Sugrue, L. P.; Seung, H. S.; and Newsome, W. T. 2005. Linear-nonlinear-Poisson models of primate choice dynamics. *Journal of the experimental analysis of behavior*, 84(3): 581–617.

Duan, Y.; Zhou, J.; Wang, Z.; Chang, Y.-C.; Wang, Y.-K.; and Lin, C.-T. 2023. Domain-specific denoising diffusion probabilistic models for brain dynamics. *arXiv preprint arXiv:2305.04200*.

Guo, M.-H.; Liu, Z.-N.; Mu, T.-J.; and Hu, S.-M. 2022. Beyond self-attention: External attention using two linear layers for visual tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5): 5436–5447.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Hung, K.-H.; Wang, K.-C.; Liu, K.-C.; Chen, W.-L.; Lu, X.; Tsao, Y.; and Lin, C.-W. 2024. MECG-E: Mamba-based ECG Enhancer for Baseline Wander Removal. *arXiv preprint arXiv:2409.18828*.

Ito, T.; and Murray, J. D. 2022. Large-scale signal and noise correlations configure multi-task coding in human brain networks. *bioRxiv*, 2022–11.

Jiang, M.; Zeng, P.; Wang, K.; Liu, H.; Chen, W.; and Liu, H. 2023. FECAM: Frequency enhanced channel attention mechanism for time series forecasting. *Advanced Engineering Informatics*, 58: 102158.

Kafashan, M.; Jaffe, A. W.; Chettih, S. N.; Nogueira, R.; Arandia-Romero, I.; Harvey, C. D.; Moreno-Bote, R.; and Drugowitsch, J. 2021. Scaling of sensory information in large neural populations shows signatures of information-limiting correlations. *Nature communications*, 12(1): 473.

Kanitscheider, I.; Coen-Cagli, R.; and Pouget, A. 2015. Origin of information-limiting noise correlations. *Proceedings of the National Academy of Sciences*, 112(50): E6973–E6982.

Ketkar, N.; and Ketkar, N. 2017. Introduction to keras. *Deep learning with python: a hands-on introduction*, 97–111.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lai, D.; Wan, Z.; Lin, J.; Pan, L.; Ren, F.; Zhu, J.; Zhang, J.; Wang, Y.; Hao, Y.; and Xu, K. 2023. Neuronal representation of bimanual arm motor imagery in the motor cortex of a tetraplegia human, a pilot study. *Frontiers in Neuroscience*, 17: 1133928.

Liang, K.; Liu, Y.; Zhou, S.; Tu, W.; Wen, Y.; Yang, X.; Dong, X.; and Liu, X. 2023. Knowledge graph contrastive learning based on relation-symmetrical structure. *IEEE Transactions on Knowledge and Data Engineering*, 36(1): 226–238.

Liang, K.; Meng, L.; Zhou, S.; Tu, W.; Wang, S.; Liu, Y.; Liu, M.; Zhao, L.; Dong, X.; and Liu, X. 2024. MINES: Message Intercommunication for Inductive Relation Reasoning over Neighbor-Enhanced Subgraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 10645–10653.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023a. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Liu, Y.; Zhao, Z.; Xu, M.; Yu, H.; Zhu, Y.; Zhang, J.; Bu, L.; Zhang, X.; Lu, J.; Li, Y.; et al. 2023b. Decoding and synthesizing tonal language speech from brain activity. *Science Advances*, 9(23): eadh0478.

Livezey, J. A.; Bouchard, K. E.; and Chang, E. F. 2019. Deep learning as a tool for neural data analysis: speech classification and cross-frequency coupling in human sensorimotor cortex. *PLoS computational biology*, 15(9): e1007091.

Luo, D.; and Wang, X. 2024. Moderntcn: A modern pure convolution structure for general time series analysis. In *The Twelfth International Conference on Learning Representations*.

Ma, X.; Rizzoglio, F.; Bodkin, K. L.; Perreault, E.; Miller, L. E.; and Kennedy, A. 2023. Using adversarial networks to extend brain computer interface decoding accuracy over time. *elife*, 12: e84296.

Metzger, S. L.; Littlejohn, K. T.; Silva, A. B.; Moses, D. A.; Seaton, M. P.; Wang, R.; Dougherty, M. E.; Liu, J. R.; Wu,

P.; Berger, M. A.; et al. 2023. A high-performance neuroprosthesis for speech decoding and avatar control. *Nature*, 620(7976): 1037–1046.

Moreno-Bote, R.; Beck, J.; Kanitscheider, I.; Pitkow, X.; Latham, P.; and Pouget, A. 2014. Information-limiting correlations. *Nature neuroscience*, 17(10): 1410–1417.

Moses, D. A.; Metzger, S. L.; Liu, J. R.; Anumanchipalli, G. K.; Makin, J. G.; Sun, P. F.; Chartier, J.; Dougherty, M. E.; Liu, P. M.; Abrams, G. M.; et al. 2021. Neuroprosthesis for decoding speech in a paralyzed person with anarthria. *New England Journal of Medicine*, 385(3): 217–227.

Panaretos, V. M.; and Zemel, Y. 2019. Statistical aspects of Wasserstein distances. *Annual review of statistics and its application*, 6: 405–431.

Panzeri, S.; Moroni, M.; Safaai, H.; and Harvey, C. D. 2022. The structures and functions of correlations in neural population codes. *Nature Reviews Neuroscience*, 23(9): 551–567.

Qi, Y.; Liu, B.; Wang, Y.; and Pan, G. 2019. Dynamic ensemble modeling approach to nonstationary neural decoding in brain-computer interfaces. *Advances in neural information processing systems*, 32.

Qi, Y.; Zhu, X.; Xu, K.; Ren, F.; Jiang, H.; Zhu, J.; Zhang, J.; Pan, G.; and Wang, Y. 2022. Dynamic ensemble bayesian filter for robust control of a human brain-machine interface. *IEEE Transactions on Biomedical Engineering*, 69(12): 3825–3835.

Romero, F. P.; Piñol, D. C.; and Vázquez-Seisdedos, C. R. 2021. DeepFilter: an ECG baseline wander removal filter using deep learning techniques. *Biomedical Signal Processing and Control*, 70: 102992.

Sereshkeh, A. R.; Trott, R.; Bricout, A.; and Chau, T. 2017. EEG classification of covert speech using regularized neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12): 2292–2300.

Stringer, C.; Michaelos, M.; Tsyboulski, D.; Lindo, S. E.; and Pachitariu, M. 2021. High-precision coding in visual cortex. *Cell*, 184(10): 2767–2778.

Sundiang, M.; Hatsopoulos, N. G.; and MacLean, J. N. 2023. Dynamic structure of motor cortical neuron coactivity carries behaviorally relevant information. *Network Neuroscience*, 7(2): 661–678.

Uddin, L. Q. 2020. Bring the noise: reconceptualizing spontaneous neural activity. *Trends in Cognitive Sciences*, 24(9): 734–746.

Valente, M.; Pica, G.; Bondanelli, G.; Moroni, M.; Runyan, C. A.; Morcos, A. S.; Harvey, C. D.; and Panzeri, S. 2021. Correlations enhance the behavioral readout of neural population activity in association cortex. *Nature neuroscience*, 24(7): 975–986.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, J.; Kim, M.; Hernandez-Mulero, A. W.; Heitzman, D.; and Ferrari, P. 2017. Towards decoding speech production from single-trial magnetoencephalography (MEG) signals. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3036–3040. IEEE.

Willett, F. R.; Avansino, D. T.; Hochberg, L. R.; Henderson, J. M.; and Shenoy, K. V. 2021. High-performance brain-to-text communication via handwriting. *Nature*, 593(7858): 249–254.

Willett, F. R.; Kunz, E. M.; Fan, C.; Avansino, D. T.; Wilson, G. H.; Choi, E. Y.; Kamdar, F.; Glasser, M. F.; Hochberg, L. R.; Druckmann, S.; et al. 2023. A high-performance speech neuroprosthesis. *Nature*, 1–6.

Wilson, G. H.; Stavisky, S. D.; Willett, F. R.; Avansino, D. T.; Kelemen, J. N.; Hochberg, L. R.; Henderson, J. M.; Druckmann, S.; and Shenoy, K. V. 2020. Decoding spoken English from intracortical electrode arrays in dorsal precentral gyrus. *Journal of neural engineering*, 17(6): 066007.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*.

Zhang, D.; and Zhang, D. 2019. Wavelet transform. *Fundamentals of image data mining: Analysis, Features, Classification and Retrieval*, 35–44.