Computational Intelligence and Neuroscience

Spatial-Temporal Transformer-based Methods for Neural Decoding

Haonan He

School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China.

Correspondence should be addressed to Haonan He; auhaonanhe@mail.scut.edu.cn

Abstract

Neural decoding from spiking activity is an essential tool for understanding the information encoded in population neurons, especially in applications like brain-computer interface (BCI). Various quantitative methods have been proposed and have shown superiorities under different scenarios respectively. From the machine learning perspective, the decoding task is to map the high-dimensional spatial & temporal neuronal activity to the low-dimensional physical quantities (e.g., velocity, position). Because of the complex interactions and the abundant dynamics among neural circuits, good decoding algorithms usually have the capability of capturing flexible spatiotemporal structures embedded in the input feature space. Recently, the Transformer-based models are widely used in processing natural languages and images due to its superior performances in handling long-range and global dependencies. Hence, in this work we examine the potential applications of Transformers in neural decoding and introduce two Transformer-based models. Besides adapting the Transformer to neuronal data, we also propose a data augmentation method for overcoming the data shortage issue. We test our models on three experimental datasets and their performances are comparable to the previous state-of-the-art (SOTA) RNN-based methods. In addition, Transformer-based models show increased decoding performances when the input sequences are longer, while LSTM-based models deteriorate quickly. In terms of time complexity, we find our methods could be faster than recurrent architectures when dealing with long sequences. Our research suggests that Transformer-based models are important additions to the existing neural decoding solutions, especially for large datasets with long temporal dependencies.

Introduction

Neural decoding studies the relationship between neural population activities and the outside world. It is a central tool to understand how neurons encode external variables. For example, we can predict space positions based on activities of rat hippocampus [1], draw movement trajectories with activities of motor cortex [2], or predict hand gestures based on activities of somatosensory cortex[3]. It also facilitates engineering applications such as brain-computer interfaces (BCI). Rapid and accurate text typing has been achieved by decoding attempted handwriting movements from neural activity in the motor cortex[4].
From the machine learning perspective, neural decoding is to find a mapping relationship between neuronal data and physical quantities (e.g., velocity, position) observable in the

outside world. Such relationship was traditionally described by linear methods such as linear regression. Recently machine learning methods especially those based on neural networks have been widely used [5], facilitating neural decoding from various aspects.

Because of the complex interactions and the abundant dynamics among neural circuits, good decoding algorithms usually have the capability of capturing spatiotemporal structures embedded in the input feature space. Recurrent Neural Networks (RNNs) are by far the most common deep learning architectures applied in neural decoding due to its ability of dealing with sequentially dependent data [5]. Among them LSTMs are the most commonly used because they are able to learn long-range dependencies better than other recurrent structures [5–10]. Convolutional Neural Networks (CNNs) are also frequently used for decoding neural signals in the form of fMRI image, calcium image or multi-channel EEG waves [11–13] for their ability of learning local dependencies of the data. Although most of these deep learning algorithms have achieved better performances compared with traditional machine learning methods, they still suffer from problems such as gradient vanishing and the restriction to local operations.

Transformer is a neural network structure that has been widely used in machine learning community in recent years. It has achieved state-of-the-art (SOTA) performances in tasks such as natural language processing [14], object detection [15], image classification [16] and protein engineering [17], etc., suggesting its wide applicability. Such self-attention mechanism suggests Transformer's superior ability of handling long-range and global dependencies. However, Transformer-based models are still relatively little used in neural decoding.

In this work we explore applications of Transformers in neural decoding and introduce two Transformer-based models. In experimental datasets of recordings from monkey motor cortex, monkey somatosensory cortex, and rat hippocampus, our Transformer-based models achieve performances comparable to the previous SOTA RNN-based methods. Besides adapting the Transformer to spikes, we also propose a data augmentation method based on Generalized Linear Model (GLM) to generate synthetic neuronal datasets larger than real ones. We test our models on three augmented datasets and they show better decoding performances. In addition, Transformer-based models show stable and increased decoding performances when the input sequences are longer, while LSTM-based models deteriorate quickly. We analyse the time complexity of our methods and find they could be faster than recurrent architectures when input sequences are long enough. We also conduct an ablative study to investigate different components' contributions to the decoding ability of our model. Our research suggests that Transformer-based models might be an alternative in neural decoding, especially for large datasets with long temporal dependencies. Our study of the inter-neuron connectivity also provides a gist for the design of neural decoding algorithms.

Materials and Methods

2.1. Dataset

We used the same three datasets as in [5], which were separately collected from monkey motor cortex, monkey somatosensory cortex and rat hippocampus. In the task for decoding from motor cortex, monkeys moved a manipulandum that controlled a cursor on a screen [18], and we aimed to decode the x and y velocity of the cursor. The data were 21 min and contained activities of 164 neurons. The mean and median firing rates were 6.7 and 3.4 spikes/s respectively. Data were put into 50 ms bins. We used 700 ms of neural activity (the concurrent bin and 13 bins before) to predict the current movement velocity.

Dataset recorded from somatosensory cortex [19] was from the same task. It contained data of 51 min and 52 neurons. They were put into 50 ms time bins. The mean and median firing rates were 9.3 and 6.3 spikes/s respectively. We used 650 ms surrounding the movement (the concurrent bin, 6 bins before, and 6 bins after) to predict the current movement velocity. Dataset recorded from hippocampus came from the task that rats chased rewards on a platform[20,21], and we aimed to decode the x and y position of the rat. This dataset contains data over a period of 75 min from 46 neurons. They had mean and median firing rates of 1.7 and 0.2 spikes/s respectively. Data were put into 200 ms bins. We used 2 s of surrounding neural activity (the concurrent bin, 4 bins before, and 5 bins after) to predict the current position.
For all three datasets we performed the same treatment as in paper [5].

2.2. Model Structure

Spike signals are usually considered to be of spatiotemporal features. Such intuition comes from complex interactions and abundant dynamics among neural circuits. We perform a simple experiment to corroborate this idea. We build one Generalized Linear Model (GLM) [22] for each neuron respectively to get the firing rate of each time bin. We then sample it according to a Poisson distribution to generate fake spikes which do not contain much inter-neuron information. We use LSTM to decode real and synthetic datasets and the results are shown in Table 1. LSTM performs better on real datasets than on synthetic datasets, suggesting the fake spikes may have lost some features compared with the real spikes. That is because we could not decode as precisely as before when using spikes with fewer features. The lost parts are mostly spatial features. This result reminds us to pay more attention to the inter-neuron information when designing decoding algorithms.

Table 1: The LSTM performs better on real datasets than on synthetic datasets generated by GLM neurons-independently, suggesting that spikes contain spatial features.

Dataset	Somatosensory Cortex (R^2)	Motor Cortex (R^2)	Hippocampus (R^2)
Synthetic	0.7578	0.6889	0.4763
Real	**0.8621**	**0.8826**	**0.6088**

Transformer was originally used in Natural Language Processing (NLP) to take a sentence as the input of the network, where each word is embedded as a token. A spike sequence is like a sentence, and each time bin is a word. When decoding spikes, intuitively we take a spike sequence as the input and embed each time bin as a token, which is called Temporal Attention.
To fully extract the spatial features and improve decoding performances, we embed each neuron as a token so that self-attention can focus on the interactions between neurons, which is called Spatial Attention. Transformer using Spatial Attention is denoted as Spatial Transformer. We compare the different decoding ability of Spatial Transformer and Temporal Transformer. We also connect Spatial Transformer and Temporal Transformer sequentially to examine the possible improvement. We test the three models on three datasets and the results are shown in Table 2. The Temporal Transformer performs better than the Spatial Transformer and the combination of the two models do improve the results.

Table 2: Attend spikes temporally, spatially or both sequentially. Temporal Transformer extracts more features than Spatial Transformer and connecting these two models can improve the decoding results.

Model	Somatosensory	Motor Cortex	Hippocampus

	Cortex (R^2)	(R^2)	(R^2)
Temporal	0.7904	0.7762	0.5004
Spatial	0.7413	0.4601	0.0290
Spatial & Temporal	**0.8153**	**0.8208**	**0.5518**

Based on the above analyses, we modify Transformer structure and introduce a new model called Spatial-Temporal Transformer (STT). We adjust the multi-head attention in [23] by dividing it to Spatial Attention and Temporal Attention. Compared to using two Transformers sequentially, our approach reduces network parameters and information loss over layers. The structure of our model is shown in Figure 1, where two attention heads are used to extract spatial and temporal features respectively. Each attention head can be split into multiple smaller attention heads to increase the diversity of the extracted features. These heads are all simply concatenated and linearly transformed to maintain the shape the same as the input.

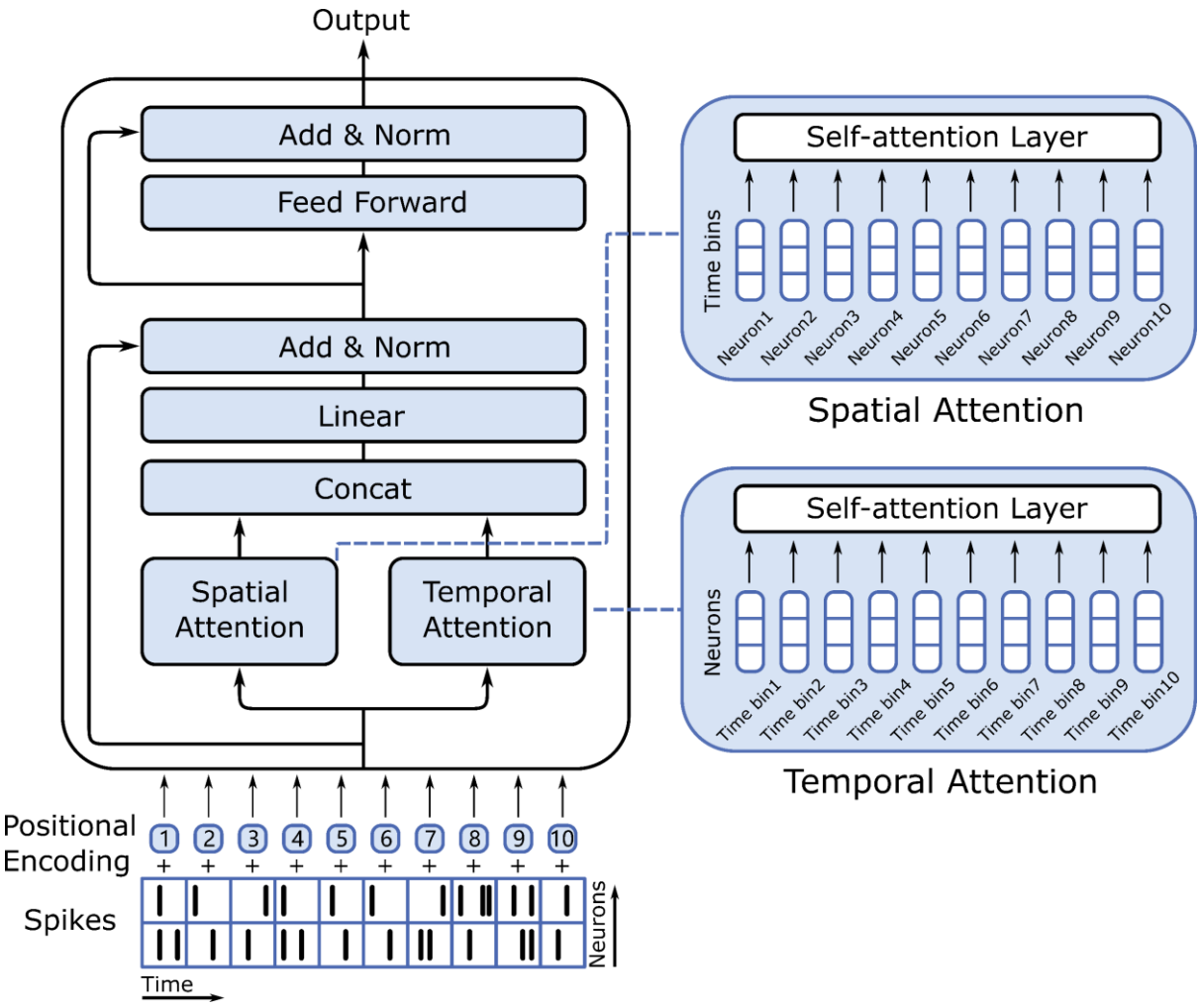

Figure 1: Spatial-Temporal Transformer (STT) architecture. Inputs are positional encoded only across time bins for their strict chronological order. A STT layer consists of spatial-temporal multi-head self-attention (STMSA) block followed by a point-wise MLP block with layer normalization (LN) and residual connections after each block.

A spike sequence $\mathbf{x} = [x_1, \ldots, x_T] \in \mathbb{R}^{T \times N}$ is added with learnable encoding embeddings $\mathbf{pos} = [\mathrm{pos}_1, \ldots, \mathrm{pos}_N] \in \mathbb{R}^{T \times N}$ to get the resulting input sequence of tokens $\mathbf{z_0} = \mathbf{x_0} + \mathbf{pos}$ of the

STT model. A STT layer consists of spatial-temporal multi-head self-attention (STMSA) block followed by a point-wise MLP block with layer normalization (LN) and residual connections after each block. For a STT model of L layers:

$$\mathbf{a_{i-1}} = \mathrm{LN}\left(\mathrm{ST\,MSA}\left(\mathbf{z_{i-1}}\right)\right) + \mathbf{z_{i-1}} \tag{1}$$

$$\mathbf{z_i} = \mathrm{LN}\left(\mathrm{MLP}\left(\mathbf{a_{i-1}}\right)\right) + \mathbf{a_{i-1}} \tag{2}$$

Where $i \in \{1,\ldots,L\}$, $\mathbf{a_i}$ is the intermediate variation between the STMSA block and the MLP block. Layer normalization of a sequence $\mathbf{x} = [x_1,\ldots,x_T]$ can be calculated as below:

$$\mathrm{LN}(\mathbf{x}) = \frac{\mathbf{x} - \mu_T}{\sqrt{\sigma_T^2 + \epsilon}} \tag{3}$$

Where μ_L and σ_T^2 are the average value and the variance of the sequence, respectively. The STMSA consists of spatial multi-head self-attention (SMSA) and temporal multi-head self-attention (TMSA). The spatial self-attention linearly maps tokens of neurons to intermediate representations, queries $\mathbf{Q_s} \in \mathbb{R}^{N \times T_q}$, keys $\mathbf{K_s} \in \mathbb{R}^{N \times T_k}$ and values $\mathbf{V_s} \in \mathbb{R}^{N \times T_v}$. The temporal self-attention maps tokens of time bins to queries $\mathbf{Q_t} \in \mathbb{R}^{T \times N_q}$, keys $\mathbf{K_t} \in \mathbb{R}^{T \times N_k}$ and values $\mathbf{V_t} \in \mathbb{R}^{T \times N_v}$. The STMSA is computed as follows:

$$\mathrm{ST\,MSA}(\mathbf{Q_s},\mathbf{Q_t},\mathbf{K_s},\mathbf{K_t},\mathbf{V_s},\mathbf{V_t}) = \mathrm{Concat}\left(\mathrm{S\,MSA}(\mathbf{Q_s},\mathbf{K_s},\mathbf{V_s}),\mathrm{T\,MSA}(\mathbf{Q_t},\mathbf{K_t},\mathbf{V_t})\right)\mathbf{W^O} \tag{4}$$

$$\mathrm{S\,MSA}(\mathbf{Q_s},\mathbf{K_s},\mathbf{V_s}) = \mathrm{softmax}\left(\frac{\mathbf{Q_s}\mathbf{K_s}^T}{\sqrt{T_k}}\right)\mathbf{V_s}^T \tag{5}$$

$$\mathrm{T\,MSA}(\mathbf{Q_t},\mathbf{K_t},\mathbf{V_t}) = \mathrm{softmax}\left(\frac{\mathbf{Q_t}\mathbf{K_t}^T}{\sqrt{N_k}}\right)\mathbf{V_t} \tag{6}$$

Where $\mathbf{W^O} \in \mathbb{R}^{2N \times N}$ represents linear transformation to keep the input shape unchanged. We scale dot-product $\mathbf{QK}^T$ with T_k and N_k to prevent the softmax function from being pushed into small gradient regions. For neural decoding, we add a linear layer to map the output of the last STT layer $\mathbf{z}_L$ to predictions of physical quantities $\mathbf{y}$.

To further enhance the feature extraction capability of our model, we combine convolutional structure with Transformer and introduce Convolutional Spatial-Temporal Transformer (CSTT) with a convolutional module before STT. The convolutional module consists of two one-dimensional convolutional layers, one is a spatial convolutional layer of size $\mathrm{N}\times 1$ and the other is a temporal convolutional layer of size 1×3. In the temporal convolutional layer, we set padding mode to 'same' to keep number of time bins unchanged. We can change convolutional kernels C to adjust the output shape of the convolutional module. The output is then fed into STT after normalization, activation and dropout layers. The architecture of CSTT is shown in Figure 3.

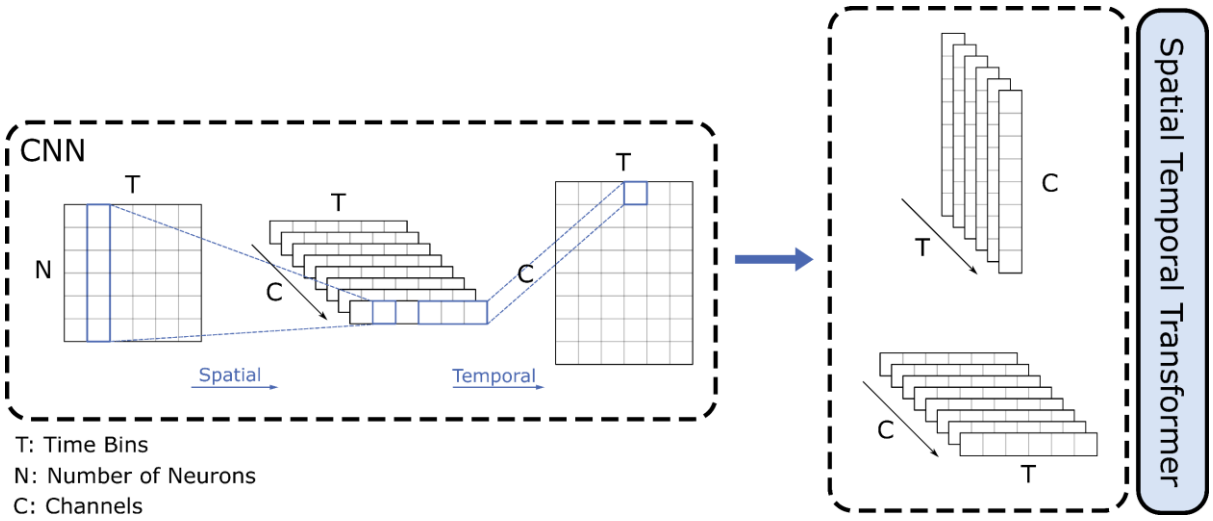

Figure 3: Convolutional Spatial-Temporal Transformer (CSTT) architecture. We add a convolutional module before STT to further improve its decoding ability. The convolutional module consists of two 1-D convolutional layers, one is a spatial convolutional layer of size $N \times 1$ and the other is a temporal convolutional layer of size 1×3. In the temporal convolutional layer, we set padding mode to 'same' to keep number of time bins unchanged. We can change convolutional kernels C to adjust the output shape of the convolutional module. The output is then fed into STT after normalization, activation and dropout layers.

The convolutional module performs initial refinement to the input. The local nature of convolutional filters can increase the diversity of features and thus benefits the decoding ability of the model.

2.3. More about attention

To get more intuitive impression on how our method works, we illustrate the attention score matrices of SMSA and TMSA in Figure 2. Attention score matrices are the softmax parts of equation 5 and 6, values of which vary from 0 to 1. They indicate the most concerned parts of the data for different attention heads.

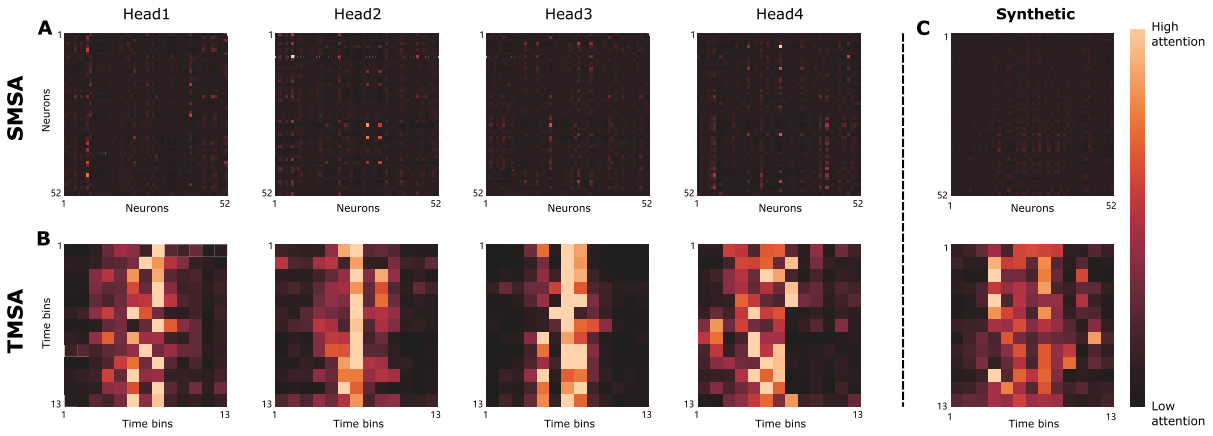

Figure 2: Attention score matrices of SMSA and TMSA. The first row represents attention scores of SMSA, the second row are results of TMSA and the fifth column (C) represents the score matrices of synthetic spikes generated by GLM. There are 8 score matrices from 4 different attention heads in A and B. Dark colours represent low attention scores and light colours represent high attention scores.

The experimental sample is collected from somatosensory cortex with 52 neurons and 13 bins. Attention heads of SMSA focus on the interactions between different neurons. Light colours in the first row of Figure 2 represent high attention scores between two neurons given by attention heads, indicating active inter-neuron interactions. SMSA could learn these latent inter-neuron features by dynamically distributing their interests to different connections between neurons. TMSA could learn time dependencies contained in spike sequences. Light colours in the second row of Figure 2 show attention heads' high interests in the relationships of target bins which contain abundant temporal features. Generally, spatial features seem to be sparser compared with temporal features, which is consistent with the results in Table 2 that Spatial Transformer performs worse than Temporal Transformer. Though with sparse spatial features, by combining inter-neuron interactions with time dependencies in STMSA, we could extract more abundant spatiotemporal features compared with traditional methods. In addition, we illustrate the attention score matrices of the synthetic spikes generated by GLM as shown in Figure 2.C. The inter-neuron attention matrix is darker than those of real spikes, suggesting the generated spikes do lose spatial features compared with real spikes.

Results

3.1. Transformer-based models are comparable to recurrent architectures

To get intuitive impression and qualitatively display the decoding ability of different decoders, we compare the decoder results with the real data from the three datasets as shown in Figure 4. The training losses are illustrated in Figure 5. We also perform quantitative measurements of the decoder results. We use $R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (y_i - \bar{y})^2}$ values to evaluate the goodness of fit, where $\hat{y}_i$ are the predicted values, y_i are the true values, and $\bar{y}$ is the mean value. We use a ten-fold cross-validation method for training, using 90% of the data as training data and 10% of the data as test data. The final R^2 value is obtained by averaging the R^2 values across the x and y components of velocity or position of the test set for each fold. The one with the highest R^2 value is considered to have the best ability of decoding. The results are shown in Table 3.

Table 3: The goodness of fit of six different models on the three datasets.

Model	Somatosensory Cortex (R^2)	Motor Cortex (R^2)	Hippocampus (R^2)
LSTM	0.8600	**0.8826**	**0.6088**
GRU	0.8592	0.8787	0.5835
CNN	0.8498	0.8399	0.5034
Spatial & Temporal	0.8153	0.8208	0.5518
STT	0.8517	0.8644	0.5828
CSTT	**0.8632**	0.8734	0.5833

Our modified Transformer-based model STT achieves higher R^2 values than Spatial & Temporal Transformer on all the three datasets, suggesting the combination of spatial and temporal attention does help improve model's decoding ability. The CSTT model performs better than STT on all three datasets due to the convolutional module. The CSTT model achieves the highest R^2 value of 0.8632 on the dataset collected from somatosensory cortex.

However, the LSTM maintains its dominance on the other two datasets. As stated in section 2.2, spatial features are comparatively sparse than temporal features in spike trains, which may explain why our STT model has no advantage over LSTM. Besides, although Transformer is known for its ability of handling long dependencies, our experimental sequences are relatively short, making the decoding performances not as good as expected.

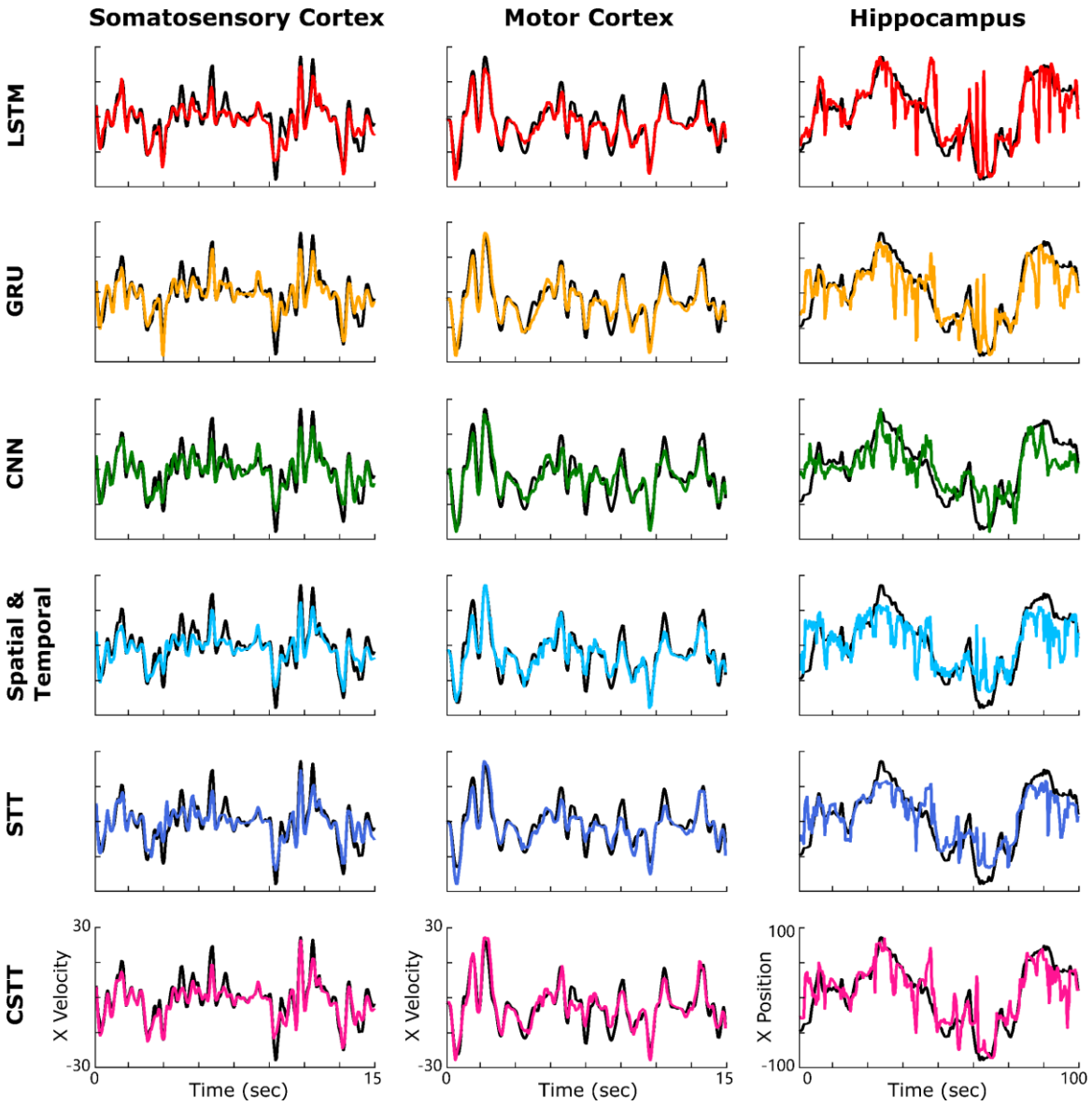

Figure 4. These are the decoder results clips, which reflect the goodness of fit. Results are from somatosensory cortex (left), motor cortex (middle) and hippocampus (right), for all 6 decoders. The black traces represent ground truth and the coloured traces are the decoder results.

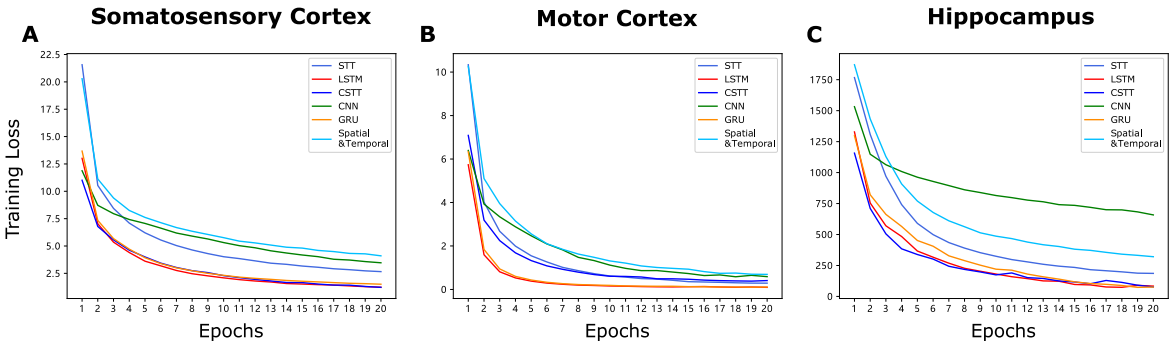

Figure 5. These are the training losses of six different models. A is the result from somatosensory cortex, B is from motor cortex and C is from hippocampus.

3.2. Transformer-based models perform better on large-scale datasets

We use GLM model described in section 2.2 to generate fake spikes and get three synthetic datasets. The volume of synthetic dataset from somatosensory cortex is ten times larger than the original with 613,390 trials. To achieve a similar size, we expand the hippocampus dataset to 30 times its original size, with 668,490 trials. For the dataset from motor cortex, we only expand it to 13 times the original size because of computational constraints. We use the three synthetic datasets to do pre-training on LSTM, STT and CSTT and then train and test with real datasets. The training process is shown in Figure 6. We compare the R^2 values of the three models before and after data augmentation and the results are shown in Table 4.

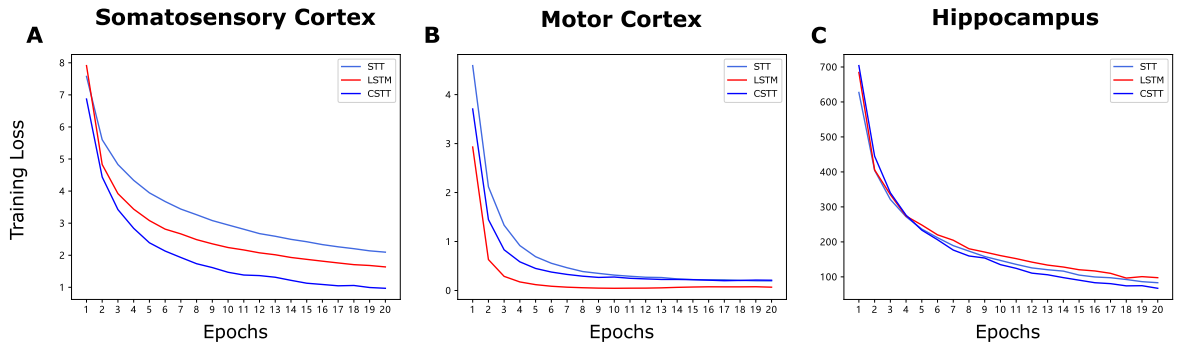

Figure 6. These are the training losses of LSTM, STT and CSTT on real datasets after pre-trained on three augmented datasets. A is the result from somatosensory cortex, B is from motor cortex and C is from hippocampus.

Table 4: Comparison of decoding ability of LSTM, STT and CSTT on the three datasets before and after augmentation.

Dataset	Model	Before (R^2)	After (R^2)
Somatosensory Cortex	LSTM	0.8600	0.8650
	STT	0.8517	0.8644
	CSTT	0.8632	**0.8734**
Motor Cortex	LSTM	0.8826	**0.8831**
	STT	0.8644	0.8711
	CSTT	0.8734	0.8828
Hippocampus	LSTM	0.6088	0.6312
	STT	0.5828	**0.6429**

	CSTT	0.5833	0.6427

After data augmentation, the performances of the three models are improved, with the two Transformer-based models showing a more significant improvement than the LSTM model. The R^2 values of STT increase by 0.0127, 0.0067 and 0.0601 on the three datasets, and those of CSTT increase by 0.0102, 0.0103 and 0.0594, while those of LSTM only increase by 0.005, 0.0005 and 0.0224. Moreover, after data augmentation, CSTT achieves the best decoding results on datasets collected from somatosensory cortex. STT scores the highest R^2 value of 0.6429 on dataset collected from hippocampus, where CSTT also performs better than LSTM. Although LSTM gets the highest R^2 value on dataset from motor cortex, the score of CSTT is quite close. Such improvements in decoding ability are consistent with our expectation that the Transformer-based models perform better on large-scale datasets. Moreover, we may achieve better results with large-scale real datasets becoming available in the future because they have more abundant spatial features than the generated synthetic datasets used in our study.

3.3. Transformer-based models can better handle long-range dependencies

In the above experiments we use 650 ms surrounding the movement (the concurrent bin, 6 bins before, and 6 bins after) for data from somatosensory cortex, 700 ms of neural activity (the concurrent bin and 13 bins before) for data from motor cortex and 2 s of surrounding neural activity (the concurrent bin, 4 bins before, and 5 bins after) for data from hippocampus. We try to increase the length of the input sequence to see how input length can influence models' decoding performances. Each time we add 10 bins to the input sequence (5 bins before and 5 bins after) and repeat 10 times. The results are shown in Figure 7.

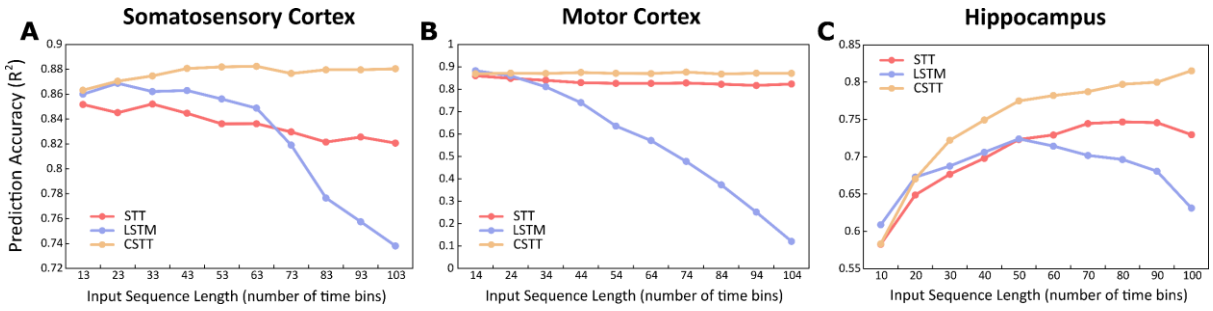

Figure 7: This is how the prediction accuracy varies with the input length. Input sequence length is increased to evaluate its effect on prediction accuracy of STT, LSTM and CSTT on datasets collected from somatosensory cortex (left), motor cortex (middle) and hippocampus (right).

The variation of prediction accuracy of the three models is generally consistent across the three datasets. On the dataset from somatosensory, the prediction accuracy of the CSTT remained between 0.86 and 0.88 as input length increases and the prediction accuracy of the STT remained stable between 0.82 and 0.86. The prediction accuracy of LSTM varies drastically with the increase of input length. As input length increases to longer than 23 bins, its prediction accuracy keeps decreasing to around 0.74 at 103 bins.
On the dataset from motor cortex, the stability difference between LSTM and the two Transformer-based models is more obvious. As the input length grows, the prediction accuracy of the LSTM decreases rapidly from nearly 0.9 at 14 bins to nearly 0.1 at 104 bins. The two Transformer-based models, however, have maintained high stability.

On the dataset from hippocampus, the prediction accuracy of LSTM increases when input length is shorter than 50 bins and then keeps deteriorating and is surpassed by STT. The prediction accuracy of the two Transformer-based models keeps growing, with STT slightly decreasing at 100 bins. CSST maintains the highest prediction accuracy after 20 bins. The high stability with the growth of sequence length shows Transformer-based models' strong ability of handling long-range dependencies.

3.4. Transformer-based models could be faster than recurrent architectures when dealing with long-dependencies

The STMSA layer consists of SMSA with complexity of $O\left(t \cdot n^2\right)$, TMSA with complexity of $O\left(t^2 \cdot n\right)$ and linear transformation with complexity of $O\left(t \cdot n^2\right)$. Its total complexity is $O\left(t^2 \cdot n + t \cdot n^2\right)$. It seems to be larger than the complexity $O\left(t \cdot n^2\right)$ of recurrent layer. Besides, with positional encoding and add & norm layers, our STT model has generally larger computational complexity than LSTM. However, the recurrent layer has the largest minimum number of sequential of $O(t)$. This is because data in recurrent layer are transmitted from cell to cell, increasing total time cost, while other types of layers can be easily parallelized because data are calculated as matrix. Per-layer complexity and minimum number of sequential operations for different layers [23] are shown in Table 5.

Table 5: Complexity and minimum number of sequential operations for different layer types. t is the sequence length, n is the representation dimension, k is the kernel size of convolutional layers and n' is the output shape of feed forward layer.

Layer Type	Complexity per Layer	Sequential Operations
STMSA	$O\left(t^2 \cdot n + t \cdot n^2\right)$	$O(1)$
Recurrent	$O\left(t \cdot n^2\right)$	$O(t)$
Convolutional	$O\left(k \cdot t \cdot n^2\right)$	$O(1)$
Positional encoding	$O(t \cdot n)$	$O(1)$
Add & norm	$O(t \cdot n)$	$O(1)$
Feed forward	$O(t \cdot n \cdot n')$	$O(1)$

According to the above analyses, though our STT model has larger computational complexity, it could achieve faster training or inference speed than LSTM when the input sequences are long enough. The CSTT model may be the slowest because of the additional convolutional layers. We randomly choose samples collected from somatosensory cortex and train and inference them with different models as shown in Table 6. We set sequences lengths to 13 bins and 63 bins to study the influence of sequential operations on total computational costs.

Table 6: Training (5 epochs) and inference time of different models on different sequence lengths.

Model	Training Time (s)		Inference Time (ms)	
	13 bins	63 bins	13 bins	63 bins
STT	16.03	**28.74**	7	**13**
LSTM	**11.85**	38.95	**5**	17
CSTT	23.65	58.63	10	26

When sequence length is relatively short of 13 bins, LSTM has the fastest training and inference speeds because it has the lowest computational complexity and the number of sequential operations is small. However, when sequence length increases to 63 bins, the number of LSTM's sequential operation increases substantially, making it much slower than STT. In general, our Transformer-based models have larger computational complexity compared with recurrent architectures but they could be calculated faster when dealing with long-range dependencies. In addition, empirically our methods could converge in a few epochs as shown in Figure 5 and 6. However, there is a lack of studies on the convergence of Transformers at present because it is quite complicated. We would like to see more researches on it in the future.

3.5. Ablative analysis

We empirically remove components of our Transformer-based model STT to evaluate their contributions to the model's decoding ability. The results are shown in Table 7.

Table 7: Contributions of different structures to the decoding ability of STT. ✓ represents the structure is contained in the model and × represents the structure is removed. The first row is the baseline.

Positional encoding	SMSA	TMSA	Layer normalize	Residual connection	R^2
✓	✓	✓	✓	✓	0.8517
×	✓	✓	✓	✓	0.5644
✓	✓	×	✓	✓	0.7413
✓	×	✓	✓	✓	0.7904
✓	✓	✓	×	✓	0.8236
✓	✓	✓	✓	×	0.8360

The positional encoding structure is crucial in our method, without which the R^2 value reduces by 0.2873 compared with the baseline. The spike trains are typical time sequences and contain abundant sequential information. The sequential information could not be learned by self-attention unless they are manually attached. This is unnecessary to recurrent models because they read sequences step by step. In addition, we find that the two types of positional encoding (sinusoidal positional encoding and learned positional encoding) work equally in our methods. Our modified self-attention layer STMSA consists of SMSA and TMSA, both contributing significantly to model's decoding ability. After removing TMSA and SMSA, the R^2 value decreases by 0.1104 and 0.0613, respectively. We also try to remove layer normalization and residual connection structures following by STMSA block and MLP block. They are not significantly important to our one-layer STT model. However, as the model gets deeper, the layer normalization and residual connection part may become momentous because of their ability of adjusting gradients.

Discussion

After all these explorations, there are still many issues remained to be explored for the application of Transformer in neural decoding:

- We use GLM to generate synthetic spikes to get large-scale datasets which lose diversity of features compared with real neuronal data. The performances of Transformer on large-scale datasets with more abundant features can be further investigated.

- Transformer's ability of dealing with long-range dependencies allows us to find appropriate input sequence length to achieve the best decoding results. Instead of manually setting the number of bins, we may let the model decide the best input length. And Transformers offer the possibility to achieve better decoding results compared with RNNs.
- We use a simple two-layer convolutional structure to form CSTT, which is proved to have better decoding ability than STT. Compared with the global feature extraction capability of Transformer, convolutional structure may have a stronger local feature extraction capability. It is worthwhile to investigate how to combine convolutional layers with Transformer better. In addition to convolutional layers, the combination of Transformer and other forms of feature extraction structures such as recurrence or graph neural network is also worth further exploration.

With the further development of neuroscience research, large-scale neuronal datasets will become available in the future. The superior processing capability of Transformer on large-scale datasets gives it potential to place recurrent model as the most widely used deep learning algorithm in the field of neural decoding. Besides, there is now a boom of research on Transformer in machine learning community. As the research on Transformer deepens, we may find models with stronger modelling ability and achieve more applications under different scenarios. Accessibility of high-volume neuronal data and research of Transformer may alter the field of neural decoding. For example, high prediction accuracy can help interpret neuron populations activities more precisely and high-speed decoding can benefit real-time applications. Hence, our research of neural decoding can help us better understand neuron circuits and facilitate engineering applications.

Conclusions

This work examines the potential applications of Transformers in neural decoding and introduces two Transformer-based models, Spatial-Temporal Transformer (STT) and Convolutional Spatial-Temporal Transformer (CSTT). We test our models on three experimental datasets and their performances are comparable to the previous SOTA RNN-based methods. We also propose a data augmentation method for overcoming the data shortage issue. By increasing the dataset volume, our models achieve significant improvement in prediction accuracy, with CSTT outperforming LSTM in all the three decoding tasks and achieving the best decoding results in datasets from somatosensory cortex. STT gets the highest R^2 value on the dataset collected from hippocampus. In addition, the STT model has stronger ability of handling long sequences and could be calculated faster compared with LSTM. Our research suggests that Transformer-based models are important additions to the existing neural decoding solutions, especially for large datasets with long-range dependencies. Our study of the inter-neuron connectivity also corroborates that neural signals contain information generated by the interactions between neurons, which provides a gist for the design of neural decoding algorithms.

Future work

We believe the future of Transformer’s application to neural signals is bright, and here we provide several research directions to be explored in the future:

1. With high-volume neuronal data becoming accessible, we could build large-scale high-capacity pre-trained models for neural decoding. Advanced brain signal collection techniques such as calcium imaging could provide much larger amount of neuronal data compared with invasive electrodes. We may use these data to train large-scale models as

universal pre-trained models. Transformer-based models are good choices for their superior performances on large-scale datasets. These models are of higher prediction accuracy compared with traditional methods and could be easily adjusted to specific tasks by using transfer learning techniques like fine-tune.

2. Transformer-based models could also be used for instant decoding tasks because of its fast inference speed. Specifically, we may replace recurrent decoding modules in BCI systems with Transformer-based models to improve their online performances. For example, RNN decoder used in brain-to-text system [4] could be replaced by our Transformer-based models. We may design smaller and faster models which could also be embedded in neurobiological data processing algorithms or compressed in BCI devices.
3. We could combine other deep learning techniques with Transformers to further enhance their capacities. We could combine them with feature extractors such as convolutional layers or new regularization such as sparsity.
4. We could customize Transformers to specific type of neuronal data such as fMRI, calcium image or EEG signals. Neuronal data are recorded in different media such as images, videos or multi-channel waves which all have different patterns. Thus, it is necessary to design specialized models for different neural activity recordings.
5. Moreover, using Transformers we could better understand how neuron populations encode physical information collaboratively. In this work, we find the existence of inter-neuron activities and examine them by visualizing attention layers in our model. We could use various deep learning techniques to decode, simulate or visualize neuron activities. It will help us to decode the secrets of the brain in the future.

Data Availability

The data that support the findings of this study are openly available and can be downloaded from
https://www.dropbox.com/sh/n4924ipcfjqc0t6/AACPWjxDKPEzQiXKUUFriFkJa?dl=0.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Funding Statement

This research received no external funding.

Acknowledgments

Thank Prof. Zhou for his guidance and encouragement.

References

[1] A Statistical Paradigm for Neural Spike Train Decoding Applied to Position Prediction from Ensemble Firing Patterns of Rat Hippocampal Place Cells | Journal of Neuroscience, (n.d.). https://www.jneurosci.org/content/18/18/7411 (accessed August 11, 2021).

[2] L. Paninski, M.R. Fellows, N.G. Hatsopoulos, J.P. Donoghue, Spatiotemporal tuning of motor cortical neurons for hand position and velocity, J Neurophysiol. 91 (2004) 515–532. https://doi.org/10.1152/jn.00587.2002.

[3] Decoding hand gestures from primary somatosensory cortex using high-density ECoG - PubMed, (n.d.). https://pubmed.ncbi.nlm.nih.gov/27926827/ (accessed August 11, 2021).
[4] High-performance brain-to-text communication via handwriting | Nature, (n.d.). https://www.nature.com/articles/s41586-021-03506-2 (accessed August 11, 2021).
[5] J.I. Glaser, A.S. Benjamin, R.H. Chowdhury, M.G. Perich, L.E. Miller, K.P. Kording, Machine Learning for Neural Decoding, ENeuro. 7 (2020) ENEURO.0506-19.2020. https://doi.org/10.1523/ENEURO.0506-19.2020.
[6] N. Ahmadi, T.G. Constandinou, C.-S. Bouganis, Decoding Hand Kinematics from Local Field Potentials Using Long Short-Term Memory (LSTM) Network, in: 2019 9th International IEEE/EMBS Conference on Neural Engineering (NER), 2019: pp. 415–419. https://doi.org/10.1109/NER.2019.8717045.
[7] J. Park, S.-P. Kim, Estimation of speed and direction of arm movements from M1 activity using a nonlinear neural decoder, in: 2019 7th International Winter Conference on Brain-Computer Interface (BCI), 2019: pp. 1–4. https://doi.org/10.1109/IWW-BCI.2019.8737305.
[8] S. Naufel, J.I. Glaser, K.P. Kording, E.J. Perreault, L.E. Miller, A muscle-activity-dependent gain between motor cortex and EMG, J Neurophysiol. 121 (2019) 61–73. https://doi.org/10.1152/jn.00329.2018.
[9] C. Heelan, J. Lee, R. O'Shea, L. Lynch, D.M. Brandman, W. Truccolo, A.V. Nurmikko, Decoding speech from spike-based neural population recordings in secondary auditory cortex of non-human primates, Commun Biol. 2 (2019) 1–12. https://doi.org/10.1038/s42003-019-0707-9.
[10] A. Du, S. Yang, W. Liu, H. Huang, Decoding ECoG Signal with Deep Learning Model Based on LSTM, in: TENCON 2018 - 2018 IEEE Region 10 Conference, 2018: pp. 0430–0435. https://doi.org/10.1109/TENCON.2018.8650433.
[11] C. Li, D.C.W. Chan, X. Yang, Y. Ke, W.-H. Yung, Prediction of Forelimb Reach Results From Motor Cortex Activities Based on Calcium Imaging and Deep Learning, Front. Cell. Neurosci. 0 (2019). https://doi.org/10.3389/fncel.2019.00088.
[12] A. Petrosyan, M. Lebedev, A. Ossadtchi, Decoding neural signals with a compact and interpretable convolutional neural network, BioRxiv. (2020) 2020.06.02.129114. https://doi.org/10.1101/2020.06.02.129114.
[13] D. Dash, P. Ferrari, J. Wang, Decoding Imagined and Spoken Phrases From Non-invasive Neural (MEG) Signals, Front Neurosci. 14 (2020) 290. https://doi.org/10.3389/fnins.2020.00290.
[14] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, ArXiv:1810.04805 [Cs]. (2019). http://arxiv.org/abs/1810.04805 (accessed July 27, 2021).
[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-End Object Detection with Transformers, ArXiv:2005.12872 [Cs]. (2020). http://arxiv.org/abs/2005.12872 (accessed July 27, 2021).
[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, ArXiv:2010.11929 [Cs]. (2021). http://arxiv.org/abs/2010.11929 (accessed July 27, 2021).
[17] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, X. Chen, J. Canny, P. Abbeel, Y.S. Song, Evaluating Protein Transfer Learning with TAPE, ArXiv:1906.08230 [Cs, q-Bio, Stat]. (2019). http://arxiv.org/abs/1906.08230 (accessed July 27, 2021).
[18] J.I. Glaser, M.G. Perich, P. Ramkumar, L.E. Miller, K.P. Kording, Population coding of conditional probability distributions in dorsal premotor cortex, Nat Commun. 9 (2018) 1788. https://doi.org/10.1038/s41467-018-04062-6.
[19] A.S. Benjamin, H.L. Fernandes, T. Tomlinson, P. Ramkumar, C. VerSteeg, R.H. Chowdhury, L.E. Miller, K.P. Kording, Modern Machine Learning as a Benchmark for Fitting Neural Responses, Front Comput Neurosci. 12 (2018) 56. https://doi.org/10.3389/fncom.2018.00056.
[20] K. Mizuseki, A. Sirota, E. Pastalkova, G. Buzsáki, Multi-unit recordings from the rat hippocampus made during open field foraging., (2009) 180 GB. https://doi.org/10.6080/K0Z60KZ9.
[21] K. Mizuseki, A. Sirota, E. Pastalkova, G. Buzsáki, Theta oscillations provide temporal windows for local circuit computation in the entorhinal-hippocampal loop, Neuron. 64 (2009) 267–280. https://doi.org/10.1016/j.neuron.2009.08.037.
[22] Point Process Generalized Linear Models, (n.d.). https://mark-kramer.github.io/Case-Studies-Python/09.html (accessed August 11, 2021).
[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, ArXiv:1706.03762 [Cs]. (2017). http://arxiv.org/abs/1706.03762 (accessed July 27, 2021).