Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Improving Brain Decoding Methods and Evaluation

Master's Thesis

Yiming Cai

yimcai@student.ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Damian Pascual, Béni Egressy
Prof. Dr. Roger Wattenhofer

August 22, 2021

Acknowledgements

I thank Beni Egressy, Damian Pascual and Oliver Richter for their kind support, constructive advice and detailed guidance throughout my master's thesis. Their input enlightened me a lot and greatly enriched my knowledge. Following their keen suggestions, I improved my skills to a great extent.

Abstract

Brain decoding is the process of inferring external stimuli from observed brain activities. Recent research has shown the possibility of decoding a fMRI scan into a vector embedding of the word that the scanned subject is reading. We argue that the vector embedding is noisy with information irrelevant to semantics, thus hindering the brain decoding performance. Therefore in this work we aim to directly classify a fMRI scan as a word within a pre-defined vocabulary, for which we propose a neural-network-based model. Besides, we consider a more realistic setup in which we train and evaluate our decoder model with data from multiple subjects, unlike most existing works merely considering the same subject in both phases. We explore various methods to improve the performance of our brain decoder model. Our complete model achieves 4.22% Top-1 and 12.87% Top-5 accuracy in the mentioned challenging setup, outperforming existing baselines. Moreover, we further validate the design of our classication-based decoder model combined with the direct classification task by testing variations to both our model and the evaluation task. In the end, we study the contribution of our data and show the potential room for improvement to our brain decoder model with extra data from different subjects.

Contents

Chapter 1

Introduction

Due to the rapid development of brain imaging techniques, it may be possible to infer people's perception from the scans of their brains, the process of which is also called brain decoding. To give a more formal definition, the brain decoding task aims at inferring the external stimuli from given brain activities. The brain decoding ability has significant applications in various fields, such as medical assistance to patients with language disabilities and consumer study which aims at understanding what customers are thinking or noticing. In related research works, brain decoding related to language always attracts attention since language plays an important role for communications between people and the external world. Some researchers like Pereira et al. [1] and Sun et al. [2] have shown the possibility of decoding the vector representations of a word read by a person from the functional Magnetic Resonance Imaging (fMRI) scans of the brain during the reading process. They show that the inferred representations of given scans tend to be more similar to the the actual vector embeddings of the corresponding words than to other words. In these works, they simply build inferential models based on ridge regression or multi-layer perceptrons (MLPs) with a heavy reliance on subject-specific data pre-processing approaches and carefully designed feature selection methods.

In this work, we use a more demanding setup, which we call direct classification, to figure out how precisely we can decode brain activities to the corresponding word stimuli. In this direct classification task, we aim at directly classifying a given fMRI scan as a word within a pre-defined vocabulary rather than comparing pairwise similarities between predictions of word representations and actual word embeddings. Moreover, we attempt to generalize brain decoding among different subjects, i.e. we use data from multiple subjects for model training and evaluation. This is a well-known difficult problem since the subjects are naturally different and thus have various numbers of voxels in their unaligned fMRI scans. On one hand this challenging setup requires a model with strong generalizability without subject-specific pre-processing. On the other hand, it also allows us to utilize much larger amount of data and train a model with higher capacity compared to studies that only focus on a specific subject.

Based on the above setup, we present a neural-network-based brain decoder model that maps fMRI scans to the corresponding word stimuli. We discard subject-specific pre-processing approaches and only align the fMRI scans of all subjects based on the external knowledge of Regions of Interest from [3]. We validate our model and explore various approaches to improve its performance in the direct classification task. Then we demonstrate the performance of our best model in both the direct classification task and the classical pairwise classification task from Pereira et al. [1]. Furthermore, we experiment with variated setup combining pairwise classification and direct classification to further validate our design choices. In the end, we further study the contribution of data from the perspective of subject and Regions of Interest.

Chapter 2

Related Works

Decoding words from records of brain activities has been an eye-catching problem among researchers since the seminal work from Mitchell et al. [4]. Recently, a wide range of research works have attempted to solve the brain decoding problem from various perspectives. As mentioned in introduction, Pereira et al. [1] presented a decoder model based on ridge regression to infer word representations of the stimuli given fMRI scans. Palutucci et al. [5] proposed a model capable of zero-shot learning, i.e. learning about unseen classes in the training phase. Besides, some researchers focus on brain decoding for different language units. Wehbe et al. [6] studied decoding methods for text passages. Sun et al. [2] explored sentence decoding with distributed representations. Moreover, some researches applied brain decoder as a tool for brain science study. For example, Just et al. [7], Huth et al, [8] and Handjaras et al. [9] mainly focus on studying how language is processed in the brain with the aid of brain decoders. Some researchers also applied brain decoder models for languages in the field of Natural Language Processing (NLP). Gauthier and Levy [10] improved Transformer [11] on NLP tasks by enhancing the model's latent representations with decoded fMRI scans. Some researchers also showed interests in decoding other forms of brain signals in addition to fMRI. Muttenthaler et al. [12] applied EEG features to tune attention weights.

Note that most related works train and evaluate the decoder model with fMRI scans from the same subject, as the misalignment of scans from different subjects hinders cross-subject evaluation. Some researchers like Van et al. [13] and Nastase et al. [14] studied this problem and proposed to solve the problem with algorithmic approaches. In our case we mainly adopt the data-driven approach to generalize brain decoding among different subjects.

Chapter 3

Dataset and Evaluation Setup

In this work, we aim to build a decoder that maps fMRI scans of brain activities to the corresponding text stimuli presented to subjects. Consequently we need relevant data and appropriate methods to evaluate the performance of such a decoder, which will be introduced in detail in this chapter.

3.1 Dataset

Our study is based on the dataset provided by Pereira et al. [1]. The dataset consists of fMRI scans of 15 different subjects. Each subject is scanned with an fMRI machine whilst reading 180 different English words. During the experiment, as is depicted in Figure 3.1, each word is presented to the subject under 3 different paradigms, which serve as supporting context to ensure that subjects are focusing on specific semantic meanings of the words while being scanned. The 3 paradigms are word clouds, sentences and images. Therefore, there are 540 fMRI scans of word stimuli in total per subject, one scan for each paradigm and word.

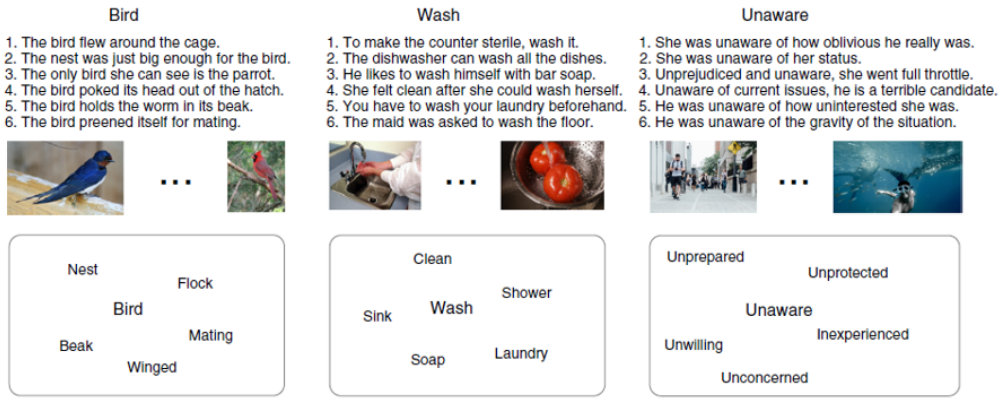

Figure 3.1: Illustration of the dataset from Pereira et al. [1]. The word *Bird* is supported with sentences, images and word clouds respectively, so are *Wash* and *Unaware*.

Moreover, 8 out of the 15 subjects took additional scans while reading 384 sentences from 96 different passages in a different dataset. 6 out of the 15 subjects also took extra scans while reading another 243 sentences from 72 passages. More details can be found in the original paper [1].

In our study we focus on text stimuli in the form of single words and thus mainly utilize scans of words in the dataset. Meanwhile, we also do some exploration on leveraging the scans of sentences as auxiliary data, e.g. using this data for pretraining the decoder model in an unsupervised manner.

3.2 Evaluation Task

As we have mentioned in the related works section, in most previous works like Pereira et al. [1], the brain decoder is designed in a regression-based manner that trains the model to generate a vector representation of the text stimuli. Then pairwise classification or similarity ranking based classification are applied to evaluate the performance of the model. Pereira et al. [1] mainly investigate the use of GloVE embeddings [15] as the prediction target and observe the pairwise classification accuracy for evaluation. As shown in Figure 3.2, for every pair of words in the dataset, they compute the cosine similarity between model predicted vectors and ground truth GloVE embedding vectors. If the similarity between the predicted vectors and the corresponding ground truth vectors is higher than the alternative ones, the classification is deemed correct. In this case, the random baseline accuracy is 50%.

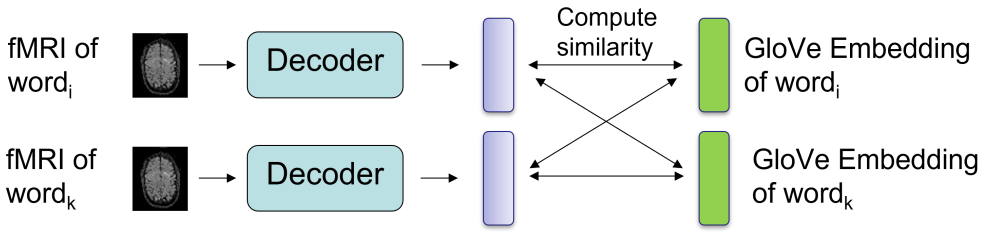

Figure 3.2: Pairwise Classification Process in Pereira et al. [1].

There are some obvious drawbacks in the above evaluation method. To begin with, GloVE vector representations also involve information irrelevant to semantics like frequency of words. Hence, the model is required to fit noisy representations for brain decoding. Furthermore, Gauthier and Ivanova's research [16] also shed light on the fact that such evaluation techniques might "fail to distinguish between representations drawn from models optimized for very different tasks". For example, even if we train the model to decode brain scans into vector representations from models for fields like machine translation and sentimental analysis, we can still obtain results close to the one in Pereira et al. [1]. Consequently we

propose to use direct classification as an alternative evaluation approach.

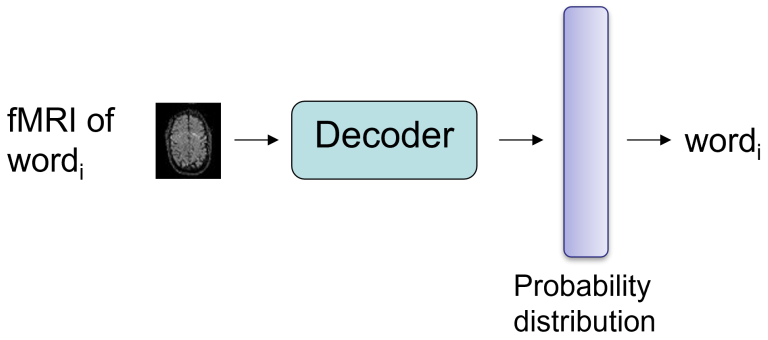

Figure 3.3: Proposed Direct Classification Approach

In direct classification, as is depicted in Figure 3.3 we build a decoder that takes fMRI scans as input and predicts the probability distribution of the word stimuli over the entire vocabulary in our dataset. In this way we can effectively figure out the exact word presented to the subject during fMRI scan. No irrelevant textual information is involved in the model. In the meantime, it is also apparently more challenging a task than pairwise classification, since the random baseline of Top-1 accuracy in this case is 1 over the vocabulary size, which is 0.6% in our case. In order to better indicate the model performance, we also report Top-5 accuracy as a referential score.

3.3 Data Splitting

Apart from the new evaluation task setup, we also consider a new data-splitting scenario. Previous works like Sun et al.[2] tend to split their data in an intra-subject way, which means training and testing with data from the same subject. This ensures data consistency in both training and testing phase. However, this is also quite an expensive approach for practical applications. In the intra-subject setting, when it comes to brain decoding for a new unseen subject, a new training set with sufficient data must be prepared so as to train a personalized brain decoder. Nevertheless, recording large amount of fMRI scans could be costly and time-consuming. According to Pereira et al. [1], it takes at least four hours to retrieve fMRI scans of 180 words per subject. Meanwhile, the limited amount of data that can be recorded for a single subject could possibly become a bottleneck in decoder model training. Consequently, designing decoder models that can work for unseen subjects without requiring a new large training set is obviously more practical and advisable.

In this work, we mainly consider a inter-subject data splitting approach following leave-one-out strategy for our evaluation. Suppose the total number of

subjects in the dataset is n, which is 15 in our case:

1. We first train the decoder model with all the fMRI scans from $n-1$ subjects
2. We perform evaluation with data from the remaining target subject.
3. In order to incorporate the advantage of the intra-subject approach, we continue to finetune the pretrained model with a certain proportion of data from the target subject.
4. Afterwards we run additional evaluation on the remaining test data from the target subject.
5. Repeat 3 and 4 in a cross-validation manner over the same target subject.

The above process is repeated over all n subjects. We perceive the leave-one-out evaluation on a certain subject as the validation experiment. Then we adjust model structure, tune hyper-parameters and run ablation study based on the validation result. Eventually the other $n-1$ leave-one-out evaluations serve as the final test.

In this setup, we simulate the practical scenario where we apply the decoder model to a new subject with only data from seen subjects for training and limited amount of new data for finetuning. Additionally, inter-subject data splitting allows us to utilize much more data than a pure intra-subject approach, which makes it possible to build models with greater capacity to learn general features among subjects. Moreover, subject-specific data pre-processing is no longer an issue for the decoder model. However, it also makes the decoding task more challenging due to the lack of alignment among fMRI scans of different subjects.

3.4 Data Alignment among Subjects

In the dataset, every single fMRI scan is stored as a 3D array of size $88 \times 128 \times 85$, which covers the entire head of the subject. However, only about 20% of the voxels contain valid information and the rest are zero padding. If we simply feed the entire array to the decoder model, the computational cost would be rather high with a low information density in the input. A common preprocessing approach is to keep only the informative voxels. According to Gorden et al. [3], the fMRI scan can be divided into different Regions of Interest (ROIs). These ROIs are associated with various functions of the brain, especially the ones related to language and perception. The dataset also provides an atlas from Gorden et al. [3] for the ROI partition. Hence, we only retrieve voxels from the ROIs as they are most relevant to our task.

In the meantime, since the subjects have different sizes of brains and ROIs, there is no spatial alignment of data across subjects. The number of voxels and spatial coordinates of voxels of the same ROI are different across subjects. Therefore, we perform zero padding to ROIs until the same ROIs across subjects have the same size. More specifically, in order to preserve more spatial information, the padding is done on each individual slice of the ROIs along the z-axis. Take the i-th ROI for instance, we first figure out the maximum number of z-axis slices n_{iz} and the highest number of voxels inside a slice n_{iv} among all subjects. Then we pad the i-th ROIs of all subjects with zero respectively until they all reach the size $n_{iv} \times n_{iz}$. In the end we flatten and concatenate all the ROIs of different sizes, transforming the fMRI scan into a vector of size 65730×1 to be the input of the decoder model. We follow the same alignment approach for all our models in the following sections.

Chapter 4

Brain Decoding Model

In order to decode text stimuli from the fMRI scans of human brain activities, we build a decoding model based on modern deep learning techniques. As shown in Figure 4.1, our complete model is implemented in the form of a classifier and mainly consists of fully connected layers.

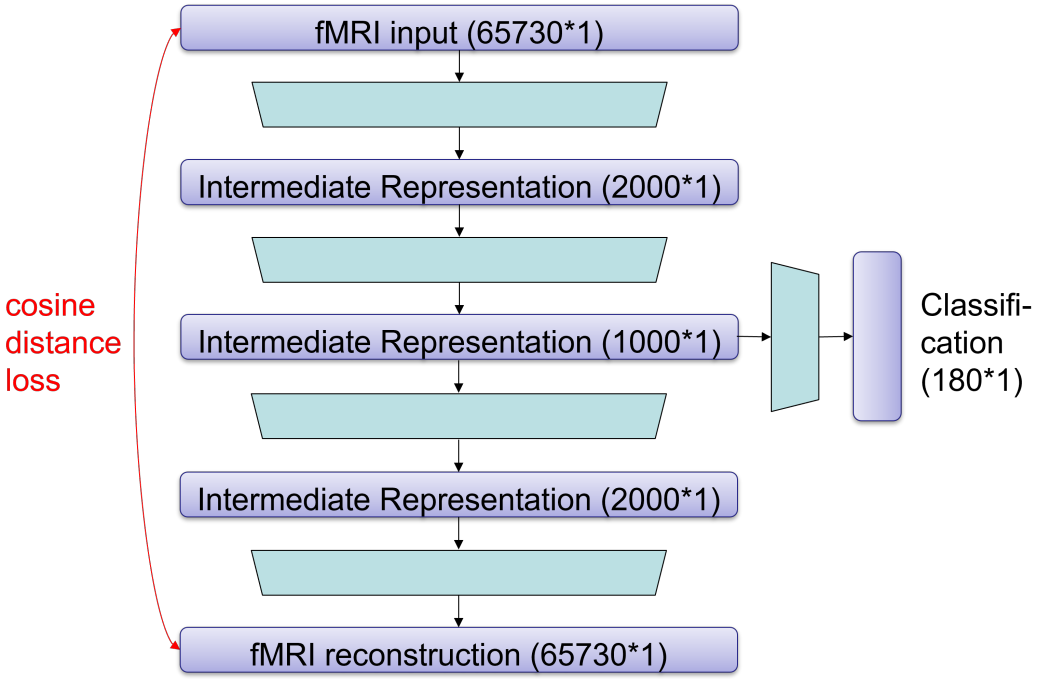

Figure 4.1: Architecture of the full model. In the figure, blue trapezoids represent fully connected layers and purple rectangles represent vectors flowing through the model. The dimensions of the vectors are shown.

As we have discussed in the Dataset and Evaluation Setup Chapter, the fMRI input data are fed into the model as vectors of size 65730×1 after alignment of

ROIs among subjects. Then two fully connected layers are applied to progressively extract intermediate representations of the fMRI input. Moreover, there are two extra fully connected layers projecting intermediate representations back to the space of higher dimension to reconstruct the fMRI input. In the end, the output layer for classification with softmax activation is aimed to produce a vector of size 180×1, which is the size of the vocabulary in our dataset. In the output vector, the i-th element o_i represents the predicted probability $y_{pred,i}$ of the fact that the i-th word in the vocabulary is the correct text stimulus. All fully connected layers except for the classification output layer are followed by batch normalization, Leaky ReLU activation (negative slope=0.3) and Dropout (rate=0.4).

During model training, We apply a cross entropy loss on the output layer, which can be computed as the following:

$$\mathcal{L}_{class} = -\sum_{i}^{v} y_{true,i} \cdot log(y_{pred,i}) \tag{4.1}$$

Where v represents vocabulary size and y_{true} is the one-hot vector representation of the ground truth word stimulus.

To begin with, we run initial experiments with a simpler baseline model, which only involves one hidden layer projecting fMRI input of size 65730×1 to intermediate representation of size 2000×1. Then the classification layer directly takes the intermediate representation as its input. Figure 4.2 illustrates the structure of baseline model.

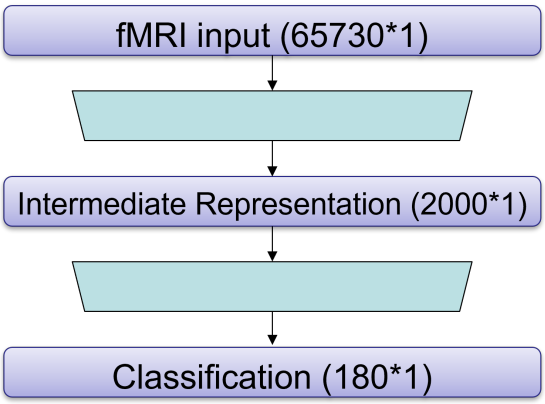

Figure 4.2: Architecture of the baseline model.

Furthermore, we apply various extensions to the baseline model so as to improve model performance. We also run an ablation study to evaluate the impact

of these extensions on model performance. Based on the experiment results, we eventually select the best model as is depicted in Figure 4.1.

4.1 FMRI Reconstruction

Since the classification is based on the latent vector extracted from fMRI input, enhancing the quality of this intermediate representation should contribute to the accuracy of classification. Therefore we apply an autoencoder structure to enhance the training signal as a regularizer. If we perceive the projection layers between fMRI input and intermediate representation as the brain decoder, then by mirroring this decoder, we can have an encoder to reconstruct the fMRI input. A reconstruction term in the following form can be added to the overall loss function:

$$\mathcal{L}_{rec} = 1 - cosine\_similarity(x_{out}, x_{in}) = \cos(x_{out}, x_{in}) \tag{4.2}$$

Where x_{out} refers to the reconstructed fMRI vector, x_{in} represents the input fMRI scan and $cos()$ is short for cosine distance.

Note that mean squared error (MSE) loss is another popular choice for autoencoder models. Besides, when implementing the autoencoder model, sharing weights between encoder and decoder is also a common approach, as it halves the required number of parameters compared to the original setup. Therefore We also conduct experiments to evaluate the effect of the above two variations. Results indicate that keeping cosine distance loss and independent weights between encoder and decoder are the best for our model. More details will be elaborated in the chapter of ablation study.

4.2 Finetuning with Additional Data

After fixing the model architecture and training, we aim to further improve model performance by finetuning the model with additional data from the target subject. In this way the model can learn some subject specific features to help with further predictions.

Since we do not have any additional dataset of fMRI scans related to word stimuli, we have to split the data from the target subject. In our setting, we finetune the model with 510 fMRI scans of 170 words, and the remaining 30 scans of 10 words are used for evaluation after finetuning. In this way, the model will not "see" data of test words from the target subject during the finetuning phase. Based on the same model checkpoint pretrained on 14 subjects, we run 18-fold cross-validation of finetuning experiments using different 10 words respectively for evaluation. In order to understand whether the finetuning approach genuinely

contributes to model performance, we report and compare both Top-1 and Top-5 accuracy of the 18 folds before and after finetuning.

Chapter 5

Ablation Study

As is mentioned before, we evaluate our models following the leave-one-out strategy and repeat the process for each subject. During our ablation study, we only use M15 as the validation subject, i.e. we train the model on the other 14 subjects and evaluate on M15 in the direct classification task. In most cases of the study we use all data from M15 for evaluation. Only when it comes to finetuning experiments do we further split the data of M15 and perform cross-validation evaluation.

5.1 Baseline Model

To begin with, we consider the baseline model which has only one fully connected layer. When training the model, we apply the Adam optimizer [17] with initial learning rate 1e-3, which further decays by factor 0.3 for every 10 epochs. We train the model for 100 epochs with early stopping monitoring the train loss and save the result. In Table 5.1, we report the evaluation result on M15. In order to account for the randomness of the initialization of the model, we run each experiment over 5 random seeds and report the mean and the standard deviation of the evaluation results. The same applies to following sections.

Model	Top-1 Acc.	Top-5 Acc.
Base	5.89% ± 0.39%	17.78% ± 0.23%

Table 5.1: Ablation Study: Result of baseline model

In our setting, the random baseline Top-1 accuracy of direct classification is 0.6%, which is about one-tenth of our baseline model. According to this result, training on different subjects did improve model performance on the validation subject, which indicates the possibility of inter-subject generalization for brain decoding.

5.2 FMRI Reconstruction

As mentioned previously, in order to enhance the intermediate representation of our model, we apply the autoencoder structure and add a cosine distance term of input and reconstructed fMRI to the loss fuction. The evaluation result is shown in Table 5.2.

Model	Top-1 Acc.	Top-5 Acc.
Base	5.89% ± 0.39%	17.78% ± 0.23%
+ Reconstruction.	6.26% ± 0.44%	17.93% ± 0.71%

Table 5.2: Ablation Study: Result of fMRI Reconstruction

Judging from the result, fMRI reconstruction effectively regularizes the training signal and improves the average Top-1 accuracy in the direct classification task. On top of that, we further explore variations of the autoencoder model as we have mentioned.

Autoencoder Variation	Top-1 Acc.	Top-5 Acc.
cos loss + share weight	5.41% ± 0.44%	15.59% ± 0.46%
cos loss + independent	**6.26% ± 0.44%**	**17.93% ± 0.71%**
mse loss + independent	5.78% ± 0.61%	16.52% ± 0.55%

Table 5.3: Experiments with variations of autoencoder

As shown in Table 5.3, we test different combinations of the reconstruction loss and autoencoder structure. When applying a cosine distance loss and an autoencoder with independent weights, we achieve both the best Top-1 and Top-5 accuracy. We argue that although sharing weights between the encoder and the decoder of the model can significantly reduce the number of model parameters and thus avoid overfitting, it simultaneously limits the flexibility of the model. Besides, the MSE loss emphasizing voxel-wise matching between input and reconstructed fMRI might also be too strict for the autoencoder, as our ultimate goal is to decode fMRI signal into word stimuli rather than perfectly reconstructing the input. Consequently, we stick to independent weights and cosine distance loss for the autoencoder extension in further tests.

5.3 Finetuning with Additional Data

In this experiment, we first train our model with 14 subjects' data following the same setting as in the fMRI reconstruction experiment. Subsequently we finetune the saved checkpoint in an 18-fold cross-validation manner with data from target subject M15. In the finetuning phase we apply another Adam optimizer [17] with

fixed learning rate 1e-5. Each finetuning experiment lasts for 20 epochs without early stopping. Complete results are shown in Table 5.4

test word	Before Finetune		After Finetune	
	Top-1 Acc.	Top-5 Acc.	Top-1 Acc.	Top-5 Acc.
0:10	0.0667	0.3333	0.1000	0.3333
10:20	0.0000	0.1000	0.1000	0.2000
20:30	0.0000	0.1000	0.0333	0.0667
30:40	0.1667	0.2333	0.1333	0.3333
40:50	0.1333	0.2667	0.1000	0.3000
50:60	0.0667	0.2000	0.1000	0.2333
60:70	0.0000	0.2000	0.1000	0.2667
70:80	0.0333	0.1000	0.0667	0.1667
80:90	0.0333	0.0667	0.0333	0.1333
90:100	0.0333	0.1667	0.1000	0.2667
100:110	0.0667	0.1667	0.1000	0.2667
110:120	0.1000	0.1333	0.0000	0.2333
120:130	0.0667	0.3333	0.1000	0.1667
130:140	0.1000	0.1333	0.1333	0.3333
140:150	0.1000	0.1667	0.0667	0.1000
150:160	0.0333	0.1000	0.0333	0.1333
160:170	0.0667	0.1333	0.0333	0.2667
170:180	0.1333	0.2333	0.1333	0.2000
arith_mean	0.0667	0.1759	0.0815	0.2222
geo_mean	0.0160	0.1600	0.0471	0.2048

Table 5.4: Ablation Study: Finetuning experiments on target subject M15

Each row of the table represents the result of a certain fold of the finetuning experiment. The first column "test word" indicates the zero-based index range (left inclusive and right exclusive) of the evaluation words in the vocabulary. The following columns report direct classification accuracy on fMRI scans of the 10 evaluation words before and after finetuning for comparison. Take the first row for instance, we use fMRI scans of first ten words in the vocabulary for evaluation and finetune the pretrained model with fMRI scans of the remaining 170 words. Before finetuning, our model achieves 6.67% Top-1 accuracy and 33.33% Top-5 accuracy on the first ten words. After finetuning Top-1 accuracy is elevated to 10.00%. The last two rows compute the arithmetic mean and geometric mean of accuracy over the 18 folds.

Judging from the results, finetuning improves direct classification accuracy in most folds of the experiment. The average accuracy over 18 folds increase significantly after finetuning, which validates our finetuning approach. In spite of the fact that data for finetuning are associated with words different from the

ones for evaluation, it still contributes to model performance by adapting the model to the target subject.

5.4 Regions of Interest

We have exploited the knowledge of Regions of Interest (ROIs) for data alignment. Since we are partitioning the fMRI scans into ROIs according to the atlas from Gordon et al [3], it is natural to consider processing each ROI with an independent small fully connected layer and then concatenate their ouputs. Compared to our original approach which processes the concatenated ROI input with a single fully connected layer, the new setting significantly reduces the number of parameters in the model. We experiment with this idea using a variated baseline model as shown in Figure 5.1.

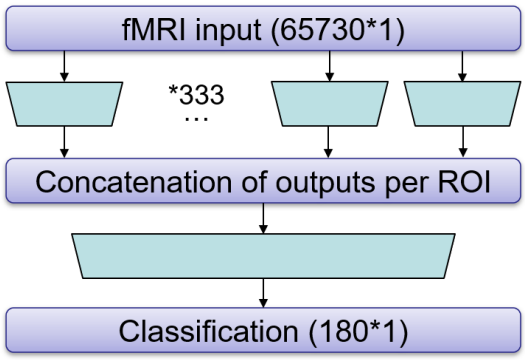

Figure 5.1: Architecture of the model with separated fully connected layers for each ROI.

Since the 333 ROIs vary in size after padding across subjects, their corresponding fully connected layers should also produce output of diffrent sizes. Here we compute the output size per ROI as the following:

$$OutputSize = max(1, \lceil \frac{InputSize}{k} \rceil) \tag{5.1}$$

Where k is a hyper-parameter to control the size of the hidden layer. We test different values of factor k and show corresponding results in Figure 5.2.

As we can see from the figure, top-1 accuracy is always lower than 5%, which is much lower than the baseline 5.89%. When the factor k is larger than 10, top-1 accuracy goes below 4%. On the other hand, even though we attempt to increase the number of model parameters by reducing the value of k to 1, i.e. producing intermediate representation of the same size as the input, the baseline model still significantly outperforms the new model. In this case, we suppose that

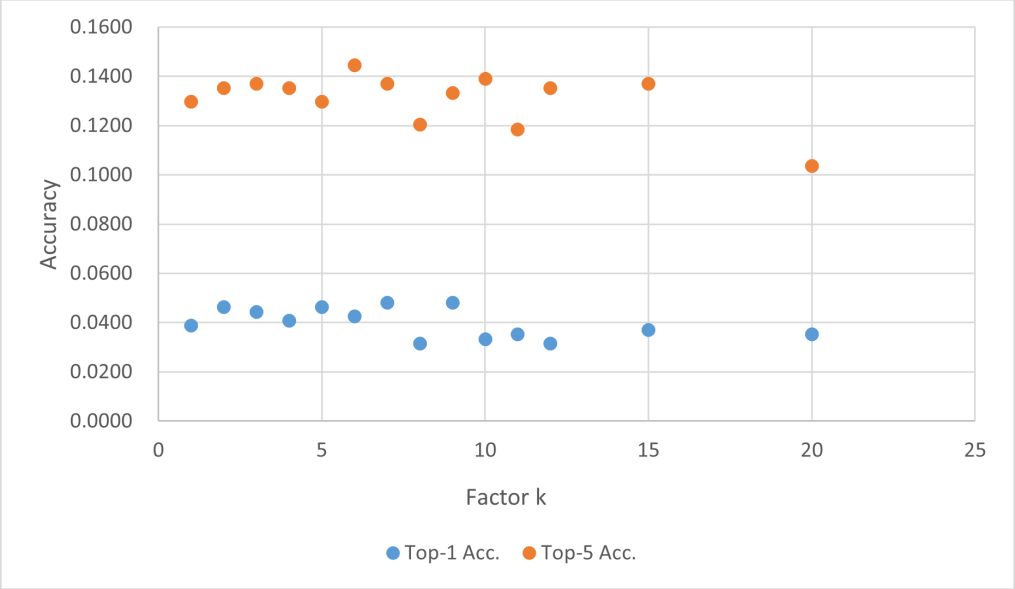

Figure 5.2: Model performance with different values of k

processing each ROI separately might result in the loss of inter-ROI information, thus having poorer performance than the baseline model. Hence, we discard this approach in further experiments.

5.5 Unsupervised Pretraining

As mentioned in dataset section, Pereira et al [1] provides additional fMRI scans of subjects reading sentences rather than words. Our model can not decode the complete sentence from a single fMRI scan. However, we can utilize this additional data exclusively on the fMRI reconstruction task. Therefore we pretrain our model in an unsupervised manner with the autoencoder extension and reconstruction loss $\mathcal{L}_{reg}$ as the only target of optimization. Then we continue to do supervised training on the word dataset with both reconstruction loss and cross entropy loss as usual. We expect the unsupervised pretraining on the sentence dataset to help the model learn general features relevant to languages, leading it to a better starting point for further supervised training. Nevertheless, as is shown in Table 5.5, the unsupervised pretraining approach does not appear to be effective. In the best result we can obtain, the average Top-5 accuracy is slightly improved compared to the case without pretraining. In the meantime, however, the average top-1 accuracy decreases to 5.96%. Considering that the unsupervised pretraining does not make significant contribution, we decide not to keep this phase in further experiments.

Model	Top-1 Acc.	Top-5 Acc.
Base	5.89% ± 0.39%	17.78% ± 0.23%
+ Reconstruction.	6.26% ± 0.44%	17.93% ± 0.71%
+ Pretraining.	5.96% ± 0.58%	18.15% ± 0.70%

Table 5.5: Ablation Study: Result with unsupervised pretraining

5.6 Mean Regularization

In order to further enhance the intermediate representation of our decoder model, we come up with mean regularization. Ideally, the model should generate the same output as well as intermediate representations for fMRI scans of the same word even if the scans are from different subjects and different paradigms. In other words, the model is supposed to focus on extracting the word exposed to the scanned subject regardless of the subject-specific physiological information. Hence, we compute the mean of the intermediate representation for each word over all subjects as the reference. Then, inspired by the triplet loss from [18], we add the following term as a regularizer to our loss function:

$$\mathcal{L}_{mean} = \sum_{i}^{v} \left(\cos\left(h_i^{(l)}, \bar{h}_i^{(l)}\right) - \sum_{j \neq i}^{v} \cos\left(h_i^{(l)}, \bar{h}_j^{(l)}\right) \right) \tag{5.2}$$

Where $\bar{h}_i^{(l)}$ is the mean of the intermediate representations for the i-th word at layer l of the model across all subjects and $h_i^{(l)}$ is the predicted intermediate representation of the i-th word at layer l.

In this study, we use mean regularization on the sole intermediate representation of the baseline model. At the beginning of the training phase, the intermediate representation retrieved by the model is not particularly informative. Therefore we initially train without mean regularization. After the training converges, we compute the mean representation for each word, and continue the training with mean regularization until early stopping occurs. Then we update the mean representation and repeat the same process iteratively.

Table 5.6 shows the results of 5 iterations of mean regularization on the baseline model. The first row marks model performance after initial training without mean regularization. After adding mean regularization, the top-5 accuracy decreases significantly, neither is the top-1 accuracy improved. Besides, due to the fluctuation of accuracy, it is difficult to determine the appropriate number of iterations. Therefore, we decide to keep the model simple and do not go further with mean regularization.

Iteration	Top-1 Acc.	Top-5 Acc.
0	5.93% ± 0.33%	17.96% ± 0.48%
1	5.41% ± 0.54%	16.48% ± 1.14%
2	5.63% ± 0.88%	16.70% ± 1.06%
3	5.78% ± 0.11%	16.85% ± 1.61%
4	5.22% ± 0.87%	16.56% ± 0.60%
5	5.96% ± 0.54%	16.67% ± 0.47%

Table 5.6: Ablation Study: Result with mean regularization

5.7 MLP-Mixer

When seeking for further improvement to the architecture of our model, we are inspired by MLP-Mixer ("Mixer" for short), a recent work from Google [19]. The Mixer is a model based exclusively on multi-layer perceptrons (MLPs) for computer vision. The model mainly consists of the following components:

1. Given an input image, split the image into S patches of the same size.
2. Project each patch to a desired dimension C with a fully connected layer and obtain the input table $\mathbf{X} \in \mathbb{R}^{S \times C}$.
3. Process $\mathbf{X}$ with Mixer layer which mainly consists of two types of MLPs. A channel-mixing MLP takes individual rows of the input table as input, and a token-mixing MLP processes each column of the input table. The detailed structure is depicted in Figure 5.3. In this way, the model retrieves information from different spatial locations and from different channels respectively.
4. Perform Global Average Pooling to the output of the Mixer layer and further projection for classification.

As we have mentioned in the Regions of Interest section, our approach processing each ROI with independent fully connected layers might lead to the loss inter-ROI information. In this case, if we treat the 333 ROIs in our input as the patches in the Mixer model, the token-mixing approach inside the Mixer architecture might potentially fix our problem since it allows communication across ROI patches. To adapt the Mixer model to our dataset and task, we need to modify the patch projector, as the ROIs in our input data vary in size. Here we propose the following two approaches to solve this problem:

1. Similar to our old approach, we use independent linear projector for each ROI and project all ROIs to the same dimension C. In this case our input table for the Mixer will be of the size $333 \times C$.

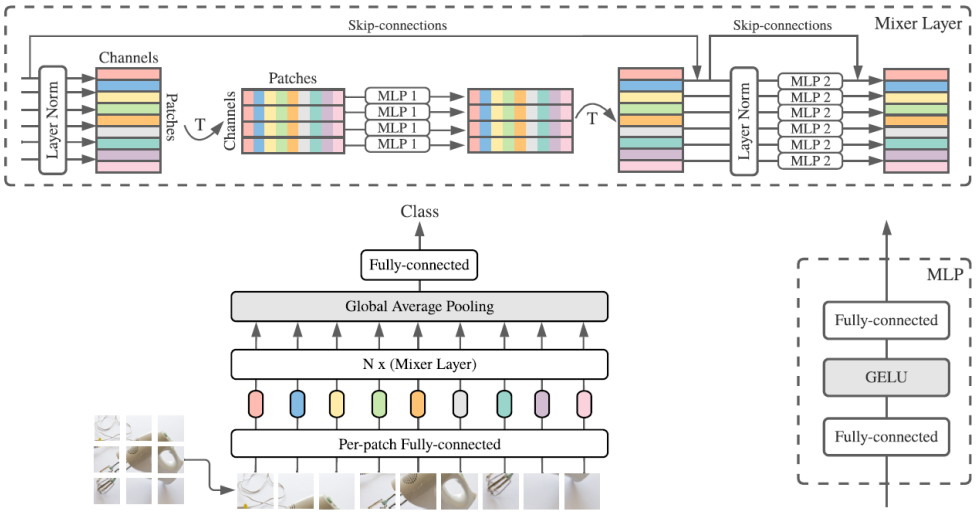

Figure 5.3: Architecture of Mixer figured in [19]

2. Similar to the original setting of the Mixer, We directly split our input vector into N patches of the same size. In this case we can apply the original patch projector in the Mixer to process each patch. After patch projection our input table will be of the size $N \times C$.

Model	Top-1 Acc.	Top-5 Acc.
Base	5.89% ± 0.39%	17.78% ± 0.23%
Mixer + Split	5.96% ± 0.22%	17.44% ± 0.86%
Mixer + ROI	<=3.30%	<= 10.56%

Table 5.7: Ablation Study: Result with MLP-Mixer. Mixer + ROI refers to our first approach that applies independent projectors per ROI. Mixer + Split represents our second approach that directly splits input vector into patches.

We show the experiment results with both two approaches in Table 5.7. When applying the first approach with independent ROI projectors, we experiment with various combinations of parameters and always have top-1 accuracy below 3.5% and top-5 accuracy below 11%, which is significant inferior to the baseline model. We further inspect the sizes of all ROIs and notice that the mean and standard deviation of ROI sizes are 197 ± 161. Therefore we assume that the huge variance in sizes of ROIs is the major limitation in this setting, as it is difficult to determine a suitable dimension for ROI projection.

Besides, with the second approach, the best result is similar to the baseline model, which is obtained with parameters $N = 1, C = 1024$. Note that the Mixer model only performs well when the number of patches N is 1, where we

actually do not split the input vector. In this case the Mixer model degenerates to stacked MLPs with Skip-Connections to some extent. As we increase the number of patches, the model performance drops significantly. Considering that we already outperform the baseline with the autoencoder extension, we decide to stay with our original model which is simpler and also effective.

Chapter 6

Results

As presented in the ablation study, both fMRI reconstruction and finetuning with additional data elevate direct classification accuracy on validation subject M15. Therefore we keep the above extensions to our model and proceed to the final test on the other 14 subjects following the same leave-one-out strategy. In addition, we also run extra experiments with various adaptations to our model in order to compare with existing works and further validate our approach.

6.1 Direct Classification

To begin with, we present the Top-1 and Top-5 direct classification accuracy of our model on all 14 test subjects in Figure 6.1 and Figure 6.2.

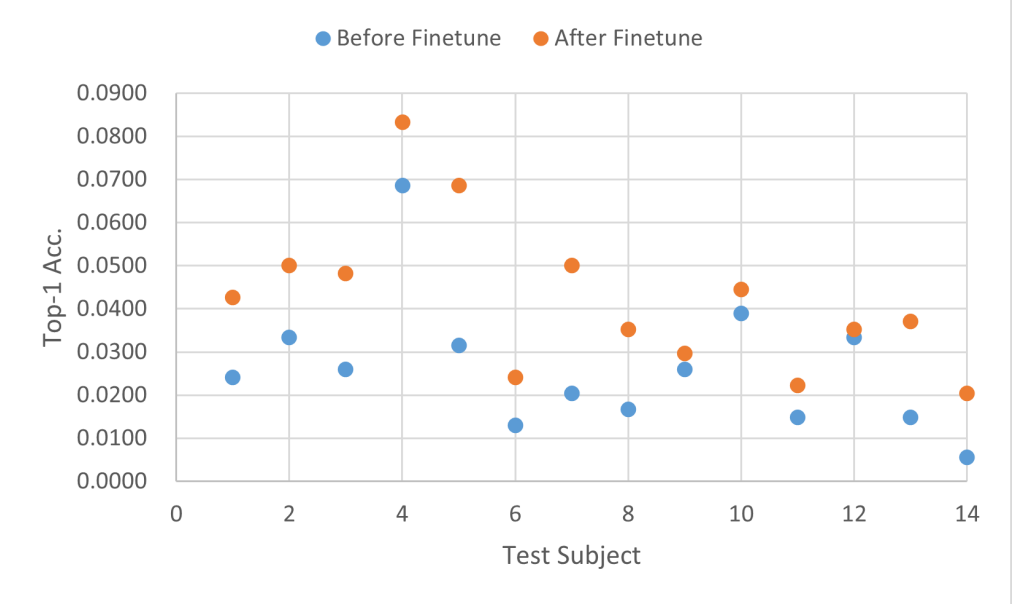

Figure 6.1: Top-1 accuracy of direct classification task on 14 test subjects

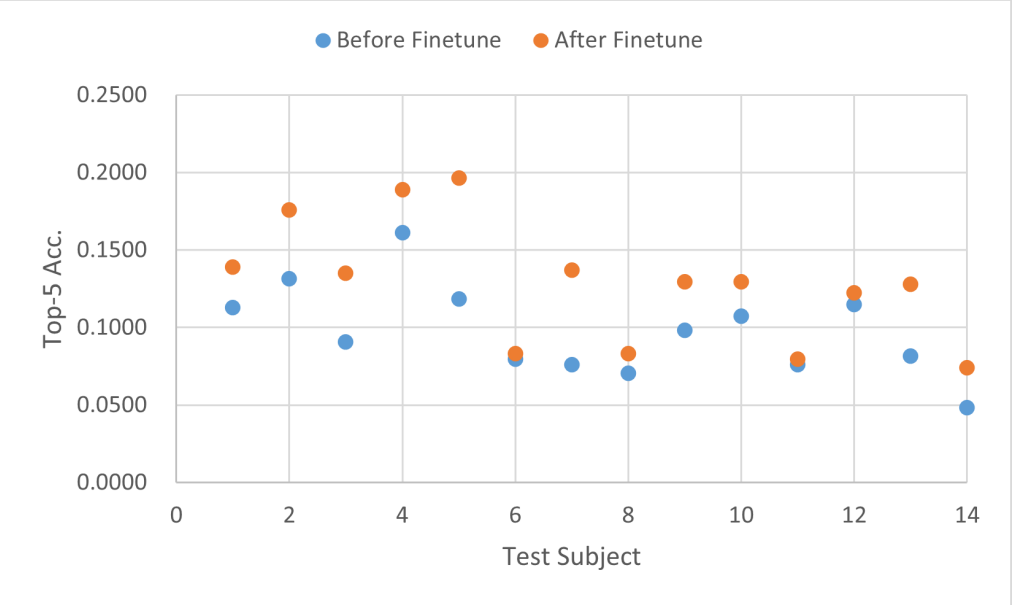

Figure 6.2: Top-5 accuracy of direct classification task on 14 test subjects

Note that each orange point in both figures represents the arithmetic mean over the results of the 18-fold finetuning experiments on a certain test subject. The blue points are the corresponding intermediate results before finetuning for comparison. We can clearly see that finetuning improves model performance for all 14 test subjects. Before finetuning, the average top-1 and top-5 accuracy over 14 tests are **2.62%** and **9.76%**, which are elevated to **4.22%** and **12.87%** respectively after finetuning.

In order to compare our model with existing works, we also consider three other baselines. We first take into account the model from Pereira et al [1], which is based on ridge regression and predicts embeddings of word stimuli. We will refer to it as the Universal Decoder in the following sections. In order to perform direct classification with Universal Decoder, we do nearest neighbour search for the model output among the GloVe embeddings of all 180 words in our vocabulary. Other settings are the same as the original work. Then we experiment with XGBoost [20], a popular regularizing gradient boosting framework performing classification based on decision trees. We apply XGBoost after performing dimensionality reduction with Principal Component Analysis (PCA). At last we evaluate the VQ-VAE model [21] which performs discrete representation learning and therefore might be capable of separating fMRI scans based on their encoded words.

As a result, in the direct classification task, the Universal Decoder achieves 0.94% average Top-1 accuracy and 4.5% average Top-5 accuracy over 14 test subjects. The performance of XGBoost with PCA is close to the random base-

line (0.6% for Top-1 and 2.8% for Top-5 accuracy), which is the worst among all compared models. The VQ-VAE model has Top-1 accuracy around 1% and Top-5 accuracy close to 5%. Note that our model significantly outperforms the others even without finetuning, which indicates that our model is more capable of generalizing to unseen subjects. Moreover, when exposed to partial data of the target subject in the finetuning phase, our model has sufficient capacity to adapt to the subject and thus boost its performance. With the good performance on the difficult but more realistic direct classification task, our model shows certain potential of applying brain decoding in a real life scenario.

6.2 Pairwise Classification

In order to further validate the adaptability of our model, we also run extra experiments that adapt neural models to the pairwise classification task from Pereira et al [1]. To be more specific, we replace the final layer for classification with a linear layer of output size 300, which is the dimension of GloVe embedding used in Pereira et al. [1]. Now that the neural models are adapted to regression-based decoders, we need a new loss function for training. In the pairwise classification task, we expect to increase the similarity between predicted output and ground-truth embedding while keeping the output away from the embedding of other words. Therefore, inspired by the triplet loss [18] again, we apply the following loss in the regression-based model and refer to it as the pairwise loss:

$$\mathcal{L}_{pw} = \sum_{i}^{v} \left(\cos\left(y_{true,i}, y_{pred,i}\right) - \sum_{j \neq i}^{v} \cos\left(y_{true,j}, y_{pred,i}\right) \right) \tag{6.1}$$

Where $y_{true,i}$ refers to the pretrained 300-dim GloVe embedding of the i-th word from [15] and $y_{pred,i}$ is the model predicted embedding of the i-th word.

We report the pairwise accuracy of our model on all 14 test subjects in Figure 6.3. Similar to the results of direct classification, finetuning also improves pairwise accuracy for our model on all 14 test subjects. The mean pairwise accuracy of our model over 14 subjects before finetuning is 70.88%, which is further improved to 74.63% by finetuning on the target subject. The Universal Decoder's accuracy is slightly lower than 70%. As for the adapted VQ-VAE, it is obviously inferior to the other two models with mean pairwise accuracy around 65%. Judging from the comparison, our model has satisfying adaptability to regression-based tasks like pairwise classification and therefore still outperforms classical methods.

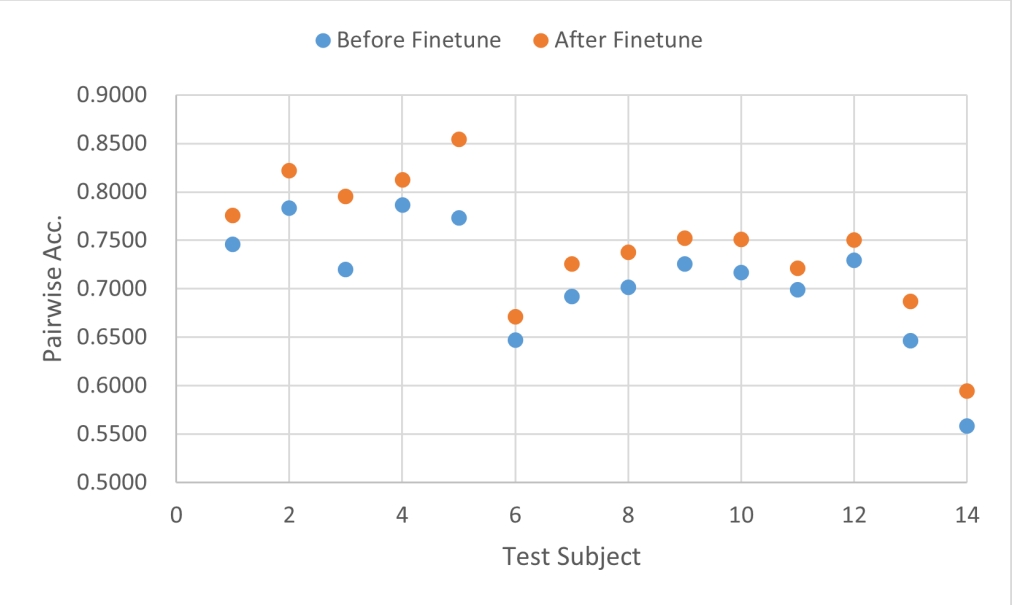

Figure 6.3: Pairwise accuracy of pairwise classification task on 14 test subjects

6.3 Variations

In this section we run additional experiments on our model with variations in the training phase and the evaluation task so as to further validate our design choices.

As mentioned above, we perform direct classification with the regression-based Universal Decoder via nearest neighbour search (NNS) in our vocabulary. This is also the exact brain decoding process with regression-based models. On the other hand this variation can also be applied to our model. In this case our model will be trained to produce vector representations and evaluated by direct classification via NNS. In order to train such a neural model, we can take into consideration the pairwise loss from the previous section or the MSE loss, a popular choice for classical regression models. We experiment with both options and present the average results over all 14 test subjects in Table 6.1.

In the table, the first row is the result of our model in default setting for direct classification task, i.e. training the model to produce a probability distribution over the vocabulary with a cross entropy loss and then doing direct classification according to the predicted probability. This result has been presented in the direct classification section of this chapter. In contrast to the default setting, the regression-based decoder always has inferior performance no matter which loss we use to train the model. Such a decoder turns out to be less accurate and effective as it has to decode brain activity indirectly relying on the word

Variation	Before Finetune		After Finetune	
	Top1-acc	Top5-acc	Top1-acc	Top5-acc
Cross Ent. + Direct clf. (Prob)	2.62%	9.76%	4.22%	12.87%
MSE + Direct clf. (NNS)	1.16%	5.34%	1.60%	6.61%
Pairwise + Direct clf. (NNS)	1.97%	8.12%	2.64%	10.29%

Table 6.1: Average results over 14 tests comparing various combinations of loss function and evaluation task on our model. Direct clf. (Prob) here refers to the direct classification based on the predicted probability as in our original model.

embedding. This validates our design choice in classification-based decoder and the evaluation task.

Besides, when training the regression-based model with the pairwise loss, the result is obviously better than the one with MSE loss, which validates the application of pairwise loss. This is expected since MSE loss requires the model to learn to fit the entire GloVe representation in all dimensions, while the pairwise loss mainly focuses on similarity among representations, which is naturally suitable for the classification task based on similarity.

Finally, putting the results of the three variations together, we can notice the trend that the more relying on GloVe embedding the setting is, the poorer performance the model tends to have. To some extent this supports our hypothesis that vector representations like GloVe are noisy. It also sheds light on the necessity of a more independent setting like our default direct classification without GloVe embedding.

Chapter 7

Contribution of Data

In this chapter, in order to further improve the performance of our model, we study the contribution of data from various perspectives. To begin with, we attempt to figure out the importance of each Region of Interest (ROI) inside the fMRI scan to the decoder model. Subsequently we compare data of the 15 subjects and study the similarities among subjects. The purpose of such studies is to help us find out and discard the least contributive data from two dimensions when training the decoder model for a specific target subject. In this way we expect to improve model performance by reducing the noises that the model learns. Furthermore, we also experiment with a simple data augmentation approach that aims to enhance the quality of our training data.

7.1 Model Attribution to Regions of Interest

We need an appropriate metric to evaluate the importance of each ROI during brain decoding. Here we adopt the data-driven approach and mainly refer to methods that attribute the prediction of a deep neural model to the input features. The quantified attribution result can be a reasonable importance metric. According to Sundararajan et al. [22], Integrated Gradients (IG) is a simple yet powerful axiomatic attribution method requiring no modification to the model. Given a deep network, the IG is defined as the path integral of the gradients along the straight-line path from a baseline input vector $x^{'}$ (zero by default) to the input vector x at hand. The Captum library [23] provides a convenient implementation of IG computation for PyTorch-based models, with which we can compute the IG of each dimension of the input. In our case, considering the high-dimensional input, we do not focus on the importance of any single feature but rather the ROI-level importance. Therefore we further compute the average IG over all dimensions of a ROI (ROI-IG) as its importance metric.

Similar to the ablation study, we use the data from M15 as the input to compute ROI-IGs with our complete model trained on the other 14 subjects. In the end we sum the ROI-IGs over all samples of M15 to get the final importance

estimations of all ROIs. As shown in Figure 7.1, some ROIs have significantly low ROI-IGs compared to others and thus are considered less contributive.

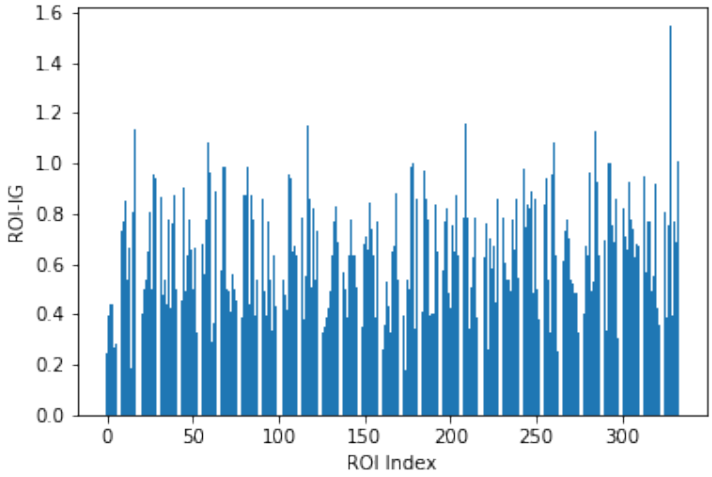

Figure 7.1: ROI-IG of the validation subject M15

Subsequently, we run a series of experiments in which we remove different numbers of the least contributive ROIs in our data based on the ROI-IG ranking, and then train and evaluate new models with input of reduced size from scratch. For comparison, we also conduct experiments in which ROIs are removed randomly. Top-1 and Top-5 accuracy of the evaluation on subject M15 are shown in Figure 7.2 and 7.3.

In general, the performance of our model is not improved no matter how many ROIs we remove in either orders. After removing more than 150 ROIs, the classification accuracy drops significantly. Note that removing ROIs randomly does not always lead to poorer performance than removing the same number of ROIs according to the importance ranking as expected. This might indicate the joint importance of ROIs with low ROI-IGs or the necessity of a more refined importance metric for our dataset. Based on the current result, we stick to using all ROIs for our model.

7.2 Reducing Training Subjects

Since we do not benefit from reducing ROIs, we consider approaches to remove the least contributive training samples to our model. As we all know, subjects are naturally different from each other. Although we have aligned the fMRI scans of different subjects to some extent, there is still much subject-specific physiological information within the data. Consequently, we aim to analyze the

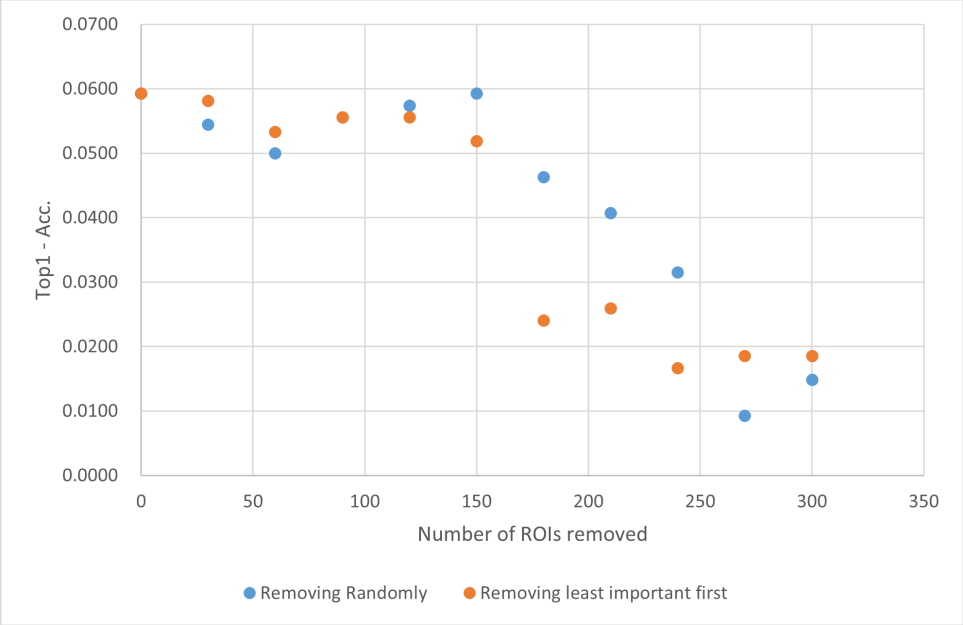

Figure 7.2: Top-1 accuracy of removing different number of ROIs according to importance

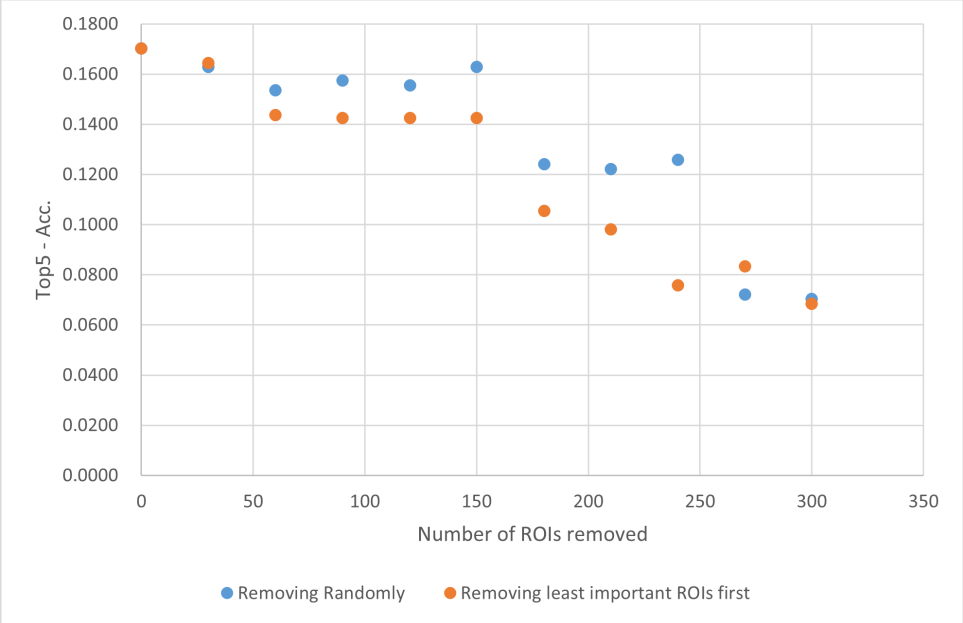

Figure 7.3: Top-5 accuracy of removing different number of ROIs according to importance

Subject	Indices of Predominant Clusters
P01	12, 1, 0
M02	6, 8, 3
M03	11, 0, 9
M04	9, 1, 0
M05	2, 3, 0
M06	7, 0, 8
M07	8, 0, 1
M08	4, 13, 10
M09	8, 0, 3
M10	0, 8, 1
M13	0, 8, 3
M14	8, 0, 1
M15	14, 1, 3
M16	0, 8, 3
M17	5, 8, 13

Table 7.1: Predominant clusters of data of each subject

similarities among subjects. With this knowledge, when given a target subject, we can exclusively use the data of the most similar subjects for decoder model training so that the model can stay more focused on language-related features.

To begin with, we perform k-means clustering with all data from the 15 subjects to check the hypothesis that there are similarities among data distribution patterns of our subjects. We naturally use 15 as the pre-defined number of clusters. According to the result, most data from the same subject falls into 1-3 particular clusters. Due to space limitation we only show the indices of 3 predominant clusters per subject in Table 7.1, which are sorted according to the number of samples in the cluster in descending order. It is easy to notice the overlaps among the predominant clusters of different subjects. Besides, some subjects with similar predominant clusters obtain similar results when used as the target subject in the direct classification task. For example, our model achieves the same top-1 accuracy 1.48% in the tests with M13 and M16 respectively. The above results support our similarity hypothesis to some extent.

To quantify the similarity between two subjects, we compute the mean of the pairwise Euclidean distances of the 540 fMRI pairs, each of which consists of two fMRI scans recorded under the same paradigm with the same word stimulus from the two target subjects respectively. Here we also use M15 as the validation subject and rank the other training subjects according to their similarities to M15. Based on this ranking we run subject removing experiments similar to the previous ROI study, i.e. we remove different numbers of subjects when training new models, and then evaluate with the data from M15. We compare three different orders of subject removing, including random order, removing the least

similar subjects first and removing the most similar subjects first. Finally, we report the Top-1 and Top-5 accuracy in Figure 7.4 and 7.5.

Figure 7.4: Top-1 accuracy on M15 of removing different numbers of training subjects

We can observe that both Top-1 and Top-5 accuracy decrease as the number of removed subjects increases. The trends of accuracy descending in all the three ways of subject removal are similar. In general removing training subjects cannot help the model improve its performance. Besides, removing the least similar subjects during training does not inevitably lead to better model performance then removing in other orders. It indicates that data from training subjects with low similarities to the target subject might still contribute to the model performance. Moreover, judging from the above results, extra data from more subjects might potentially further improve the performance of our model.

7.3 Simple Data Augmentation

Since we do not have extra data for model training, we turn to data augmentation methods. In consideration of the particular data alignment approach and our limited understanding of brain activities, we simply try an intuitive augmentation approach. Given an aligned fMRI scan x_1, we randomly sample another scan x_2 with the same word stimulus label in the training set regardless of the subject and the paradigm. Then we generate augmented data x_{new} as the following:

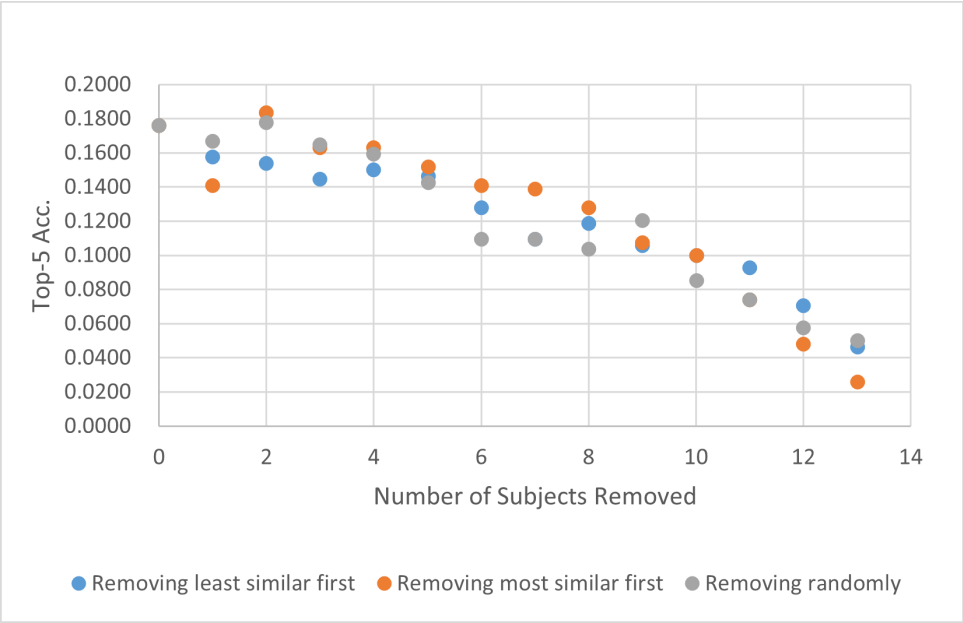

Figure 7.5: Top-5 accuracy on M15 of removing different numbers of training subjects

$$x_{new} = (1 - \epsilon)x_1 + \epsilon x_2 \tag{7.1}$$

Where $\epsilon \in [0, 1]$ is the weight of the sampled scan. We do not rule out the case where $x_1 == x_2$ since it preserves a proportion of the original input when ϵ is not zero.

We consider both fixed and random values of ϵ, train our model with the augmented data and evaluate on the validation subject M15. Results are shown in Table 7.2. When using random ϵ, we achieve 5.96% top-1 accuracy, which is better than using other fixed values. However it fails to outperform the default setting without data augmentation ($\epsilon = 0$). Therefore we do not keep this simple augmentation approach. On the other hand, since the performance of our model also does not decrease significantly with the processed data, the linear combinations of ROIs possibly can still preserve major features related to the word stimuli. On top of that, more carefully designed data augmentation approaches to extract these features and denosie might potentially further improve the current brain decoding method.

ϵ	Top-1 Acc.	Top-5 Acc.
0	6.26% ± 0.44%	17.93% ± 0.71%
0.3	5.78% ± 0.61%	17.89% ± 0.83%
0.5	5.30% ± 0.42%	17.37% ± 1.04%
0.7	5.52% ± 0.54%	16.89% ± 0.82%
random	5.96% ± 0.38%	17.59% ± 0.61%

Table 7.2: Results of data augmentation with different weights ϵ

Chapter 8

Conclusions

In this work we have presented a neural-network-based brain decoder model that maps fMRI scans to the corresponding word stimuli. Furthermore, in order to improve the brain decoder model, we explore a few methods among which fMRI reconstruction and target subject finetuning show positive results. On top of that, our model outperforms existing work with 74.63% pairwise accuracy in the pairwise classification task as well as 4.22% Top-1 and 12.87% Top-5 accuracy in the direct classification task. Moreover, we further justify the design of classication-based decoder combined with direct classification task for evaluation, as our model in this setting outperforms all variated regression-based alternatives, proving to be more efficient and accurate for brain decoding. In the end, we study the contribution of data from the perspectives of subject and Regions of Interest, which shows the potential room for improvement to our brain decoder method with extra data from different subjects.

Bibliography

[1] F. Pereira, B. Lou, B. Pritchett, S. Ritter, S. J. Gershman, N. Kanwisher, M. Botvinick, and E. Fedorenko, "Toward a universal decoder of linguistic meaning from brain activation," *Nature communications*, vol. 9, no. 1, pp. 1–13, 2018.

[2] J. Sun, S. Wang, J. Zhang, and C. Zong, "Towards sentence-level brain decoding with distributed representations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 7047–7054.

[3] E. M. Gordon, T. O. Laumann, B. Adeyemo, J. F. Huckins, W. M. Kelley, and S. E. Petersen, "Generation and evaluation of a cortical area parcellation from resting-state correlations," *Cerebral cortex*, vol. 26, no. 1, pp. 288–303, 2016.

[4] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.-M. Chang, V. L. Malave, R. A. Mason, and M. A. Just, "Predicting human brain activity associated with the meanings of nouns," *science*, vol. 320, no. 5880, pp. 1191–1195, 2008.

[5] M. M. Palatucci, D. A. Pomerleau, G. E. Hinton, and T. Mitchell, "Zero-shot learning with semantic output codes," 2009.

[6] L. Wehbe, B. Murphy, P. Talukdar, A. Fyshe, A. Ramdas, and T. Mitchell, "Simultaneously uncovering the patterns of brain regions involved in different story reading subprocesses," *PloS one*, vol. 9, no. 11, p. e112575, 2014.

[7] M. A. Just, V. L. Cherkassky, S. Aryal, and T. M. Mitchell, "A neurosemantic theory of concrete noun representation based on the underlying brain codes," *PloS one*, vol. 5, no. 1, p. e8622, 2010.

[8] A. G. Huth, W. A. De Heer, T. L. Griffiths, F. E. Theunissen, and J. L. Gallant, "Natural speech reveals the semantic maps that tile human cerebral cortex," *Nature*, vol. 532, no. 7600, pp. 453–458, 2016.

[9] G. Handjaras, E. Ricciardi, A. Leo, A. Lenci, L. Cecchetti, M. Cosottini, G. Marotta, and P. Pietrini, "How concepts are encoded in the human brain: a modality independent, category-based cortical organization of semantic knowledge," *Neuroimage*, vol. 135, pp. 232–242, 2016.

[10] J. Gauthier and R. Levy, "Linking artificial and human neural representations of language," *arXiv preprint arXiv:1910.01244*, 2019.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[12] L. Muttenthaler, N. Hollenstein, and M. Barrett, "Human brain activity for machine attention," *arXiv preprint arXiv:2006.05113*, 2020.

[13] C. E. Van Uden, S. A. Nastase, A. C. Connolly, M. Feilong, I. Hansen, M. I. Gobbini, and J. V. Haxby, "Modeling semantic encoding in a common neural representational space," *Frontiers in neuroscience*, vol. 12, p. 437, 2018.

[14] S. A. Nastase, Y.-F. Liu, H. Hillman, K. A. Norman, and U. Hasson, "Leveraging shared connectivity to aggregate heterogeneous datasets into a common response space," *NeuroImage*, vol. 217, p. 116865, 2020.

[15] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[16] J. Gauthier and A. Ivanova, "Does the brain represent words," *An evaluation of brain decoding studies of language understanding. arXiv preprint arXiv:1806.00591*, 2018.

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

[18] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[19] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, D. Keysers, J. Uszkoreit, M. Lucic *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *arXiv preprint arXiv:2105.01601*, 2021.

[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[21] A. v. d. Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," *arXiv preprint arXiv:1711.00937*, 2017.

[22] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3319–3328.

[23] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan *et al.*, "Captum: A unified and generic model interpretability library for pytorch," *arXiv preprint arXiv:2009.07896*, 2020.