Brain signals modelling while watching movies

Hardik Mittal, Gaurav Bhole

Process

1. Got the cleaned and Shared Response Model fMRI data from the dataset
2. Dropped the extra TRs as asked by the authors of the paper and the dataset
3. Used the top 1000, 5000, 10000 voxels from the dataset by choosing the ones with the highest variance across the timestamps
4. Further preprocessed the fMRI data by normalizing it and aligning it with the video data by convolving it using an HRF delay of 4 seconds.

Process

1. Got the video embeddings for the different movies by using 3 different video encoders - ViViT, ViTMAE, VideoMAE and stored the embeddings to be used for later tasks
2. Dropped the extra parts of the video from the start and the end
3. Preprocessed the video embeddings by normalizing them.
4. Built two models - one deep MLP, one LSTM based model to make encoding and decoding models from predicting the brain fMRI and the video embeddings consecutively
5. Ran multiple experiments for hyperparameter finetuning and finding different analysis
6. Made a classification model, which given, the brain fMRI, predicts the movie which is being watched by the participant

Task

1. Given the video embeddings per second (using 8 frames per second) from the video and the fMRI data with TR (repetition time) as 1.

   Use this data to build

   - Encoding models to predict the brain activity using the stimulus features
   - Decoding models to predict the stimulus features using the brain activity

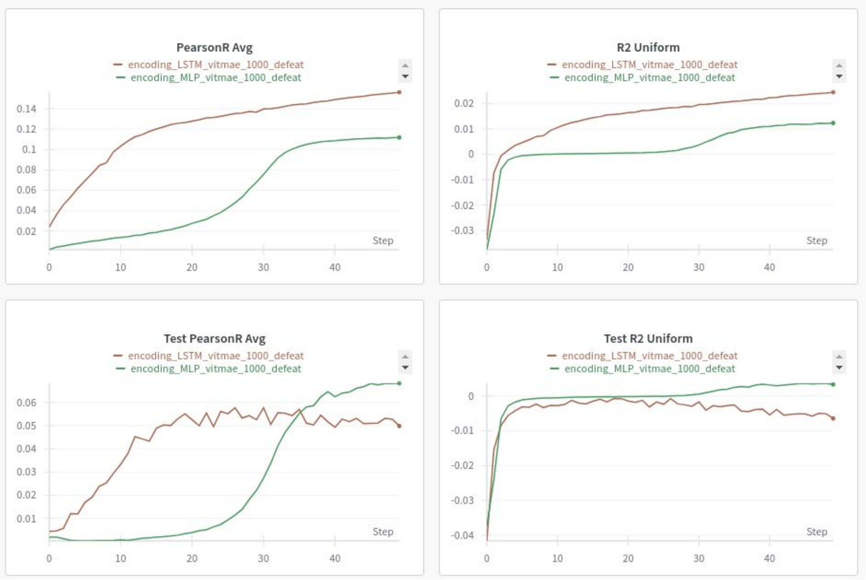
2. Use the fMRI data to predict which movie is being seen by the participant while watching it
   - Inter-subjects
   - Intra-subjects

Encoding Experiments

Across different encoder models

We tried two models

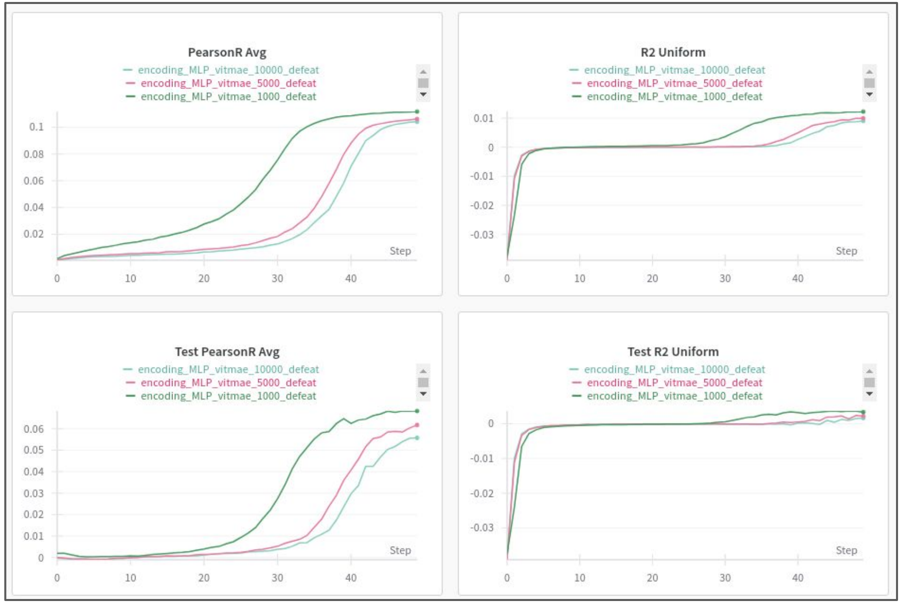
- A Deep Linear network
- An LSTM network

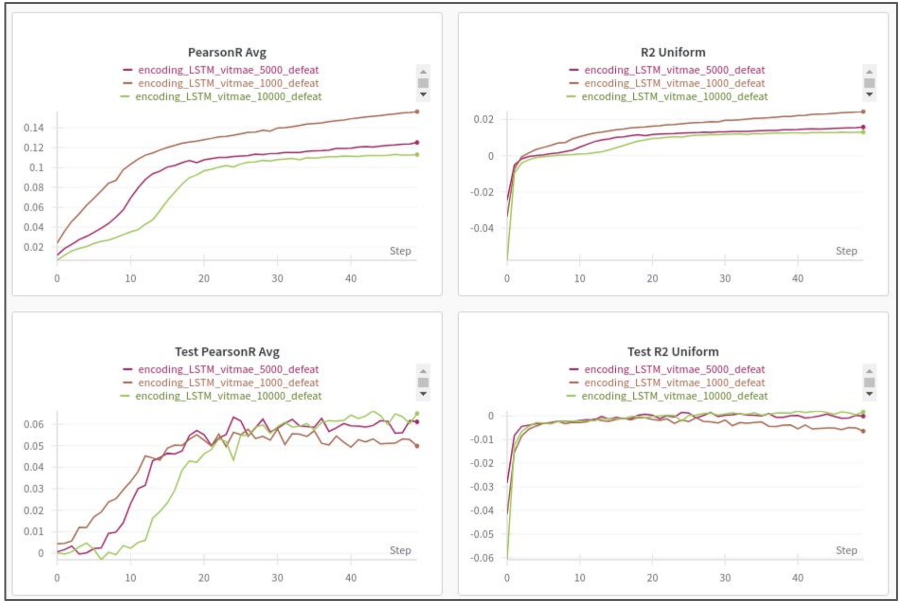


Analysis

- As we can see the LSTM network is overfitting on the train set leading to high scores on the train set but very low scores with high variance on the test set.
- Thus we selected an MLP for further thorough analysis

Across different number of active brain voxels

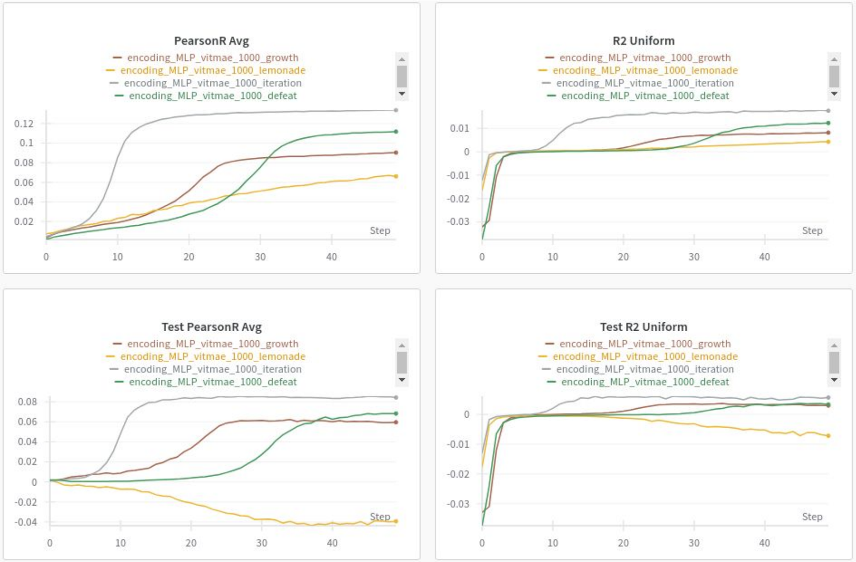




As we can see that both for MLP network and the LSTM network, the top 1000 variance based voxels perform better than having top 5000 or top 10000 voxels (found by using the max variance across the timestamps in the brain voxels)

Across different movies

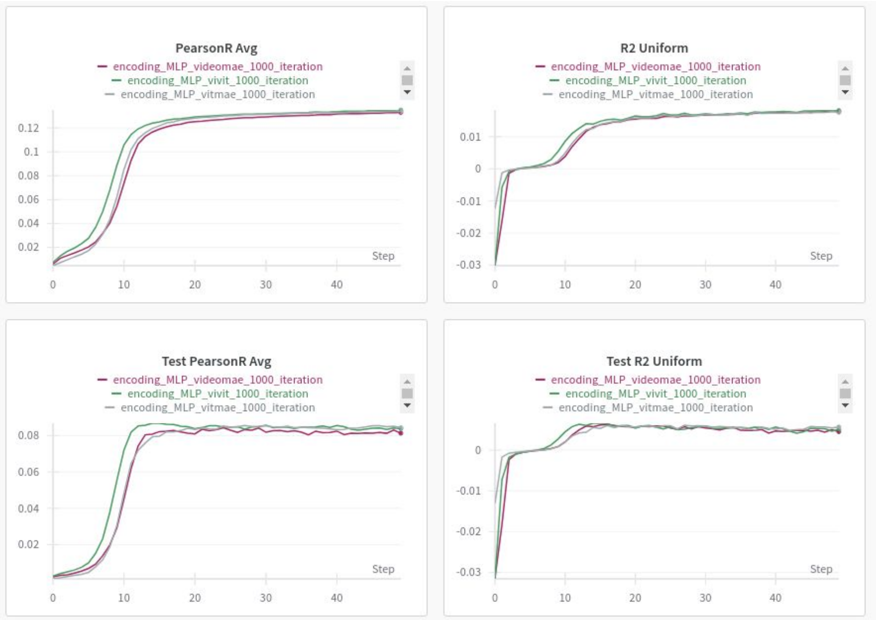
We ran all the encoding models across the different movies to encode the movie embeddings to predict the fMRI data to see which movie was giving better results and generalization across subjects

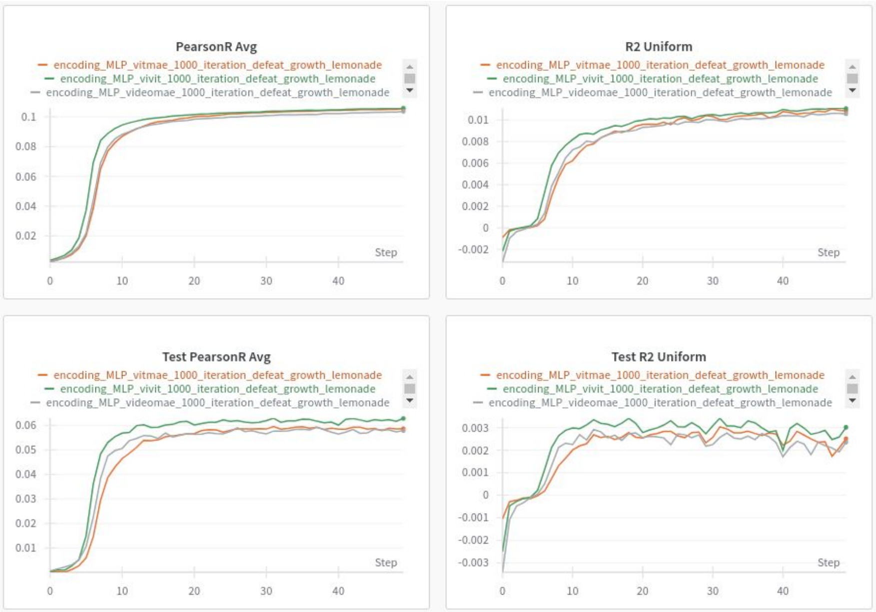


Analysis

Across all the video encoders, the movie **iteration** gave the best generalization and performance, followed by an almost equal performance by the movies, **growth & defeat** with the movie **lemonade** giving the worst performance, consistently.

Across different video encoders





As we can see, there was not much difference in the performance across the different video encoders while encoding on a single movie or on multiple movies either

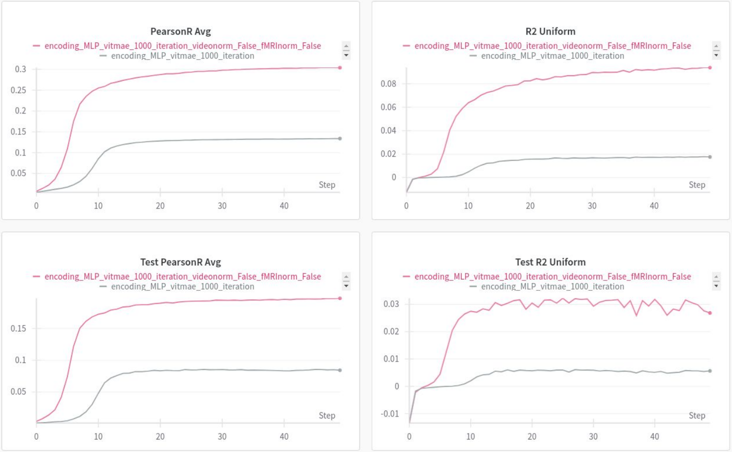
With and Without Normalization

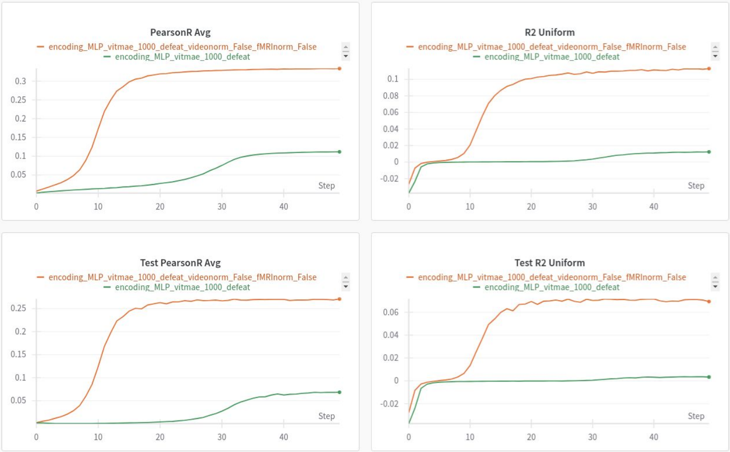
Earlier we were running all the models by doing Z-Score normalisation across all the brain voxels and the video embeddings

As asked by sir, we ran this experiment by removing the normalization for both of them

Analysis

- As we can see that, just by removing normalization while keeping everything else the exact same, we saw a major boost from 0.1 to 0.3 pearson correlation coefficient in training and 0.05 to 0.15 pearson correlation in testing, along with a similar boost in the R2 score as well.





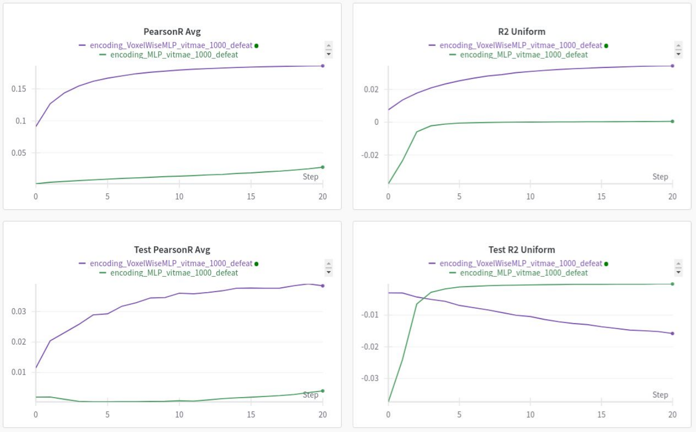
Full MLP vs Voxel-Wise model

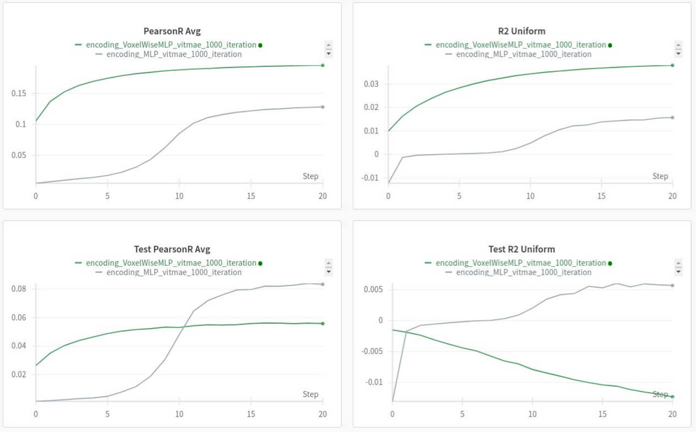
Earlier we were running all the models by doing Z-Score normalisation across all the brain voxels and the video embeddings

As asked by sir, we ran this experiment by removing the normalization for both of them

Analysis

- As we can see that, just by removing normalization while keeping everything else the exact same, we saw a major boost from 0.1 to 0.3 pearson correlation coefficient in training and 0.05 to 0.15 pearson correlation in testing, along with a similar boost in the R2 score as well.



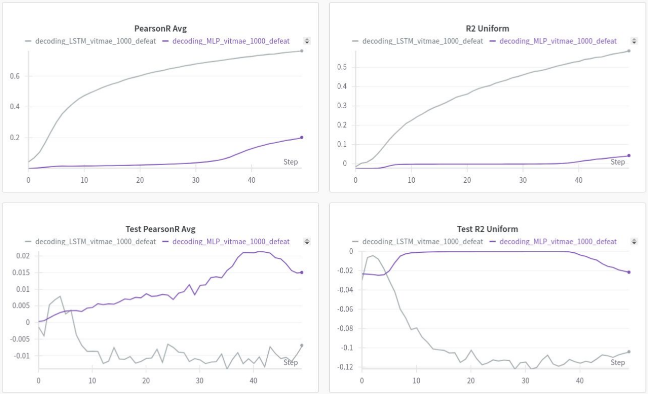


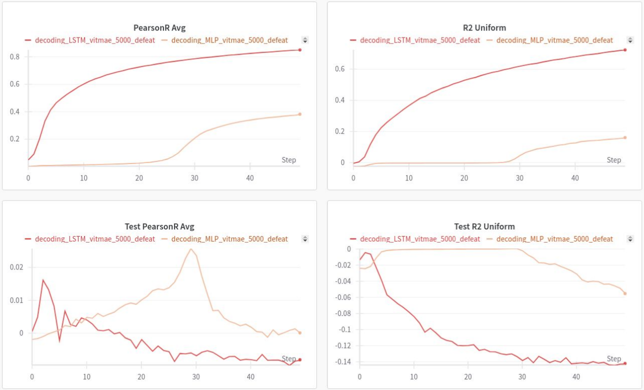
Decoding Experiments

Across different decoder models

We tried two models

- A Deep Linear network
- An LSTM network

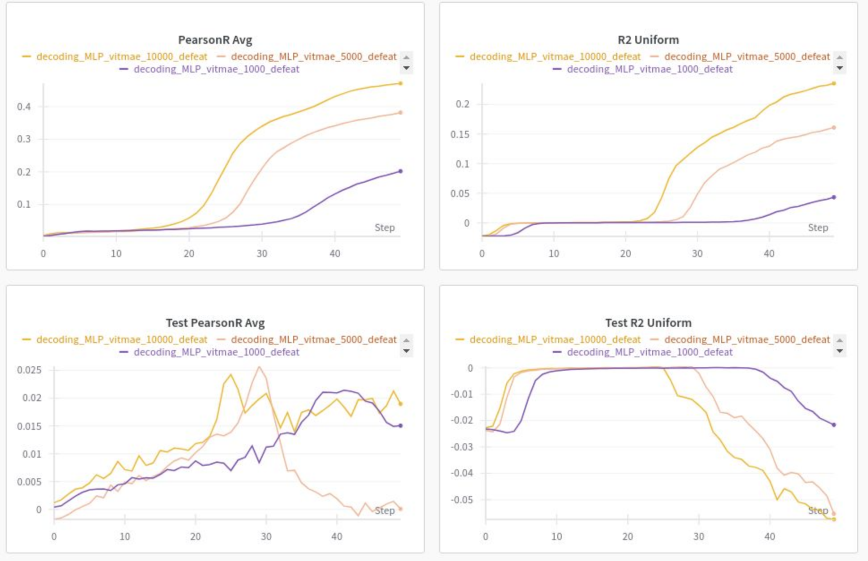


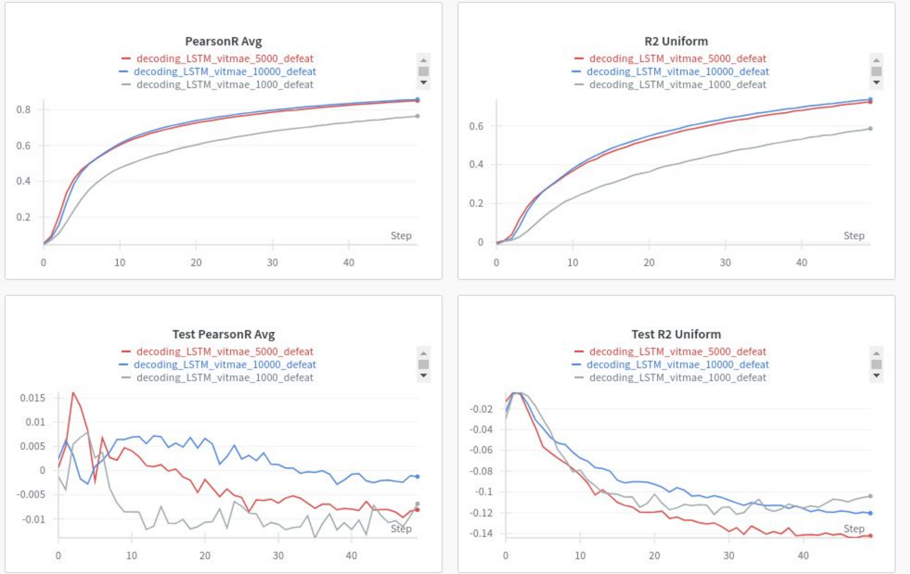


Analysis

- Just like encoding, we can see the LSTM network is overfitting on the train set leading to high scores on the train set but very low scores with high variance on the test set.

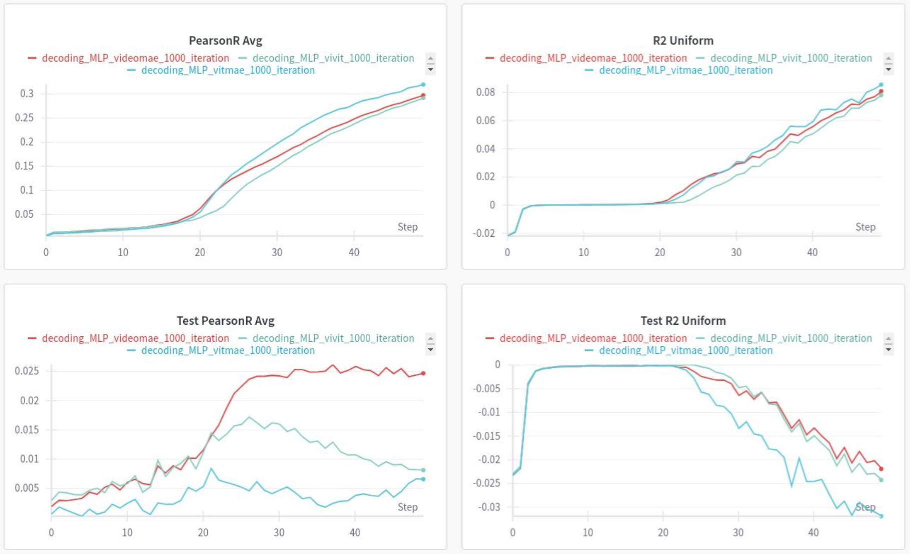
Across different number of active brain voxels

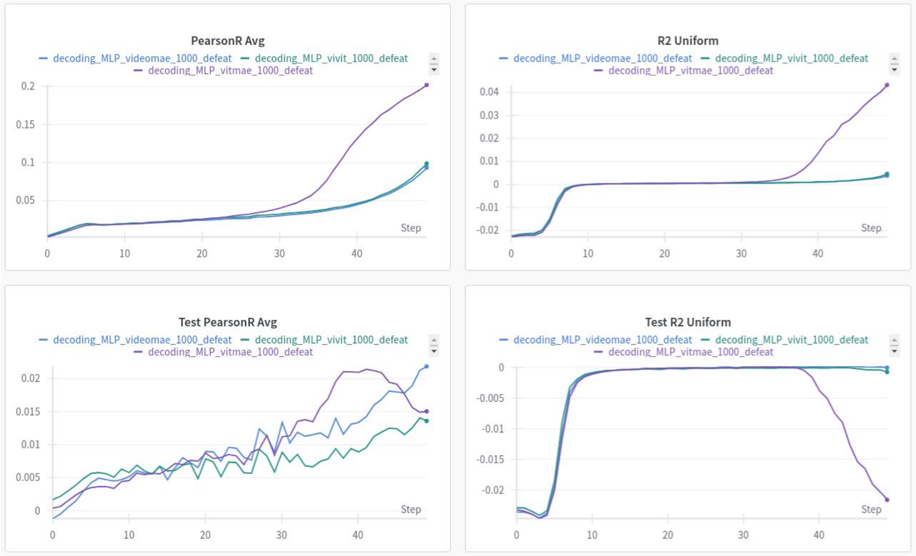




As we can see that both for MLP network and the LSTM network, unlike encoding, the **top10000** variance based voxels perform better than having **top1000** or **top5000** voxels on both the test and train set (found by using the max variance across the timestamps in the brain voxels)

Across different video encoders

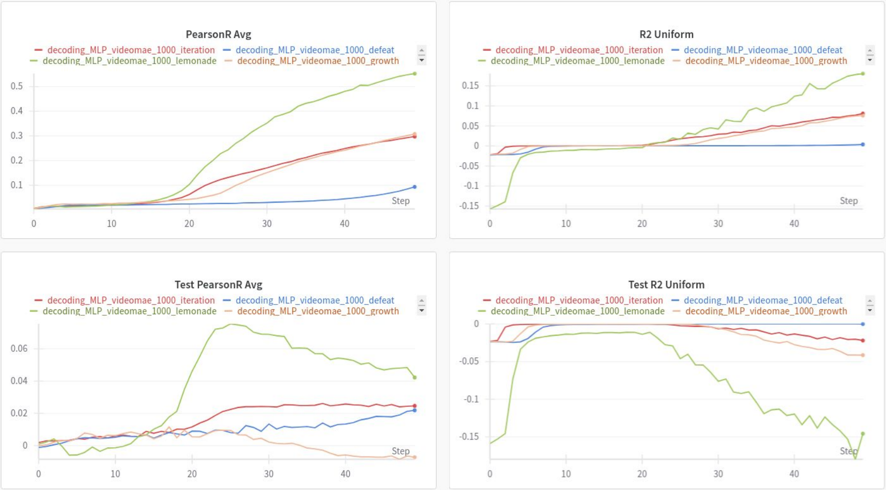




As we can see, here **vitmae** overfit on the train data leading to a higher performance there but low on the test, while **videomae** performed decently well, both on the train as well as test data

Across different movies

We ran all the decoding models across the different movies to decode the movie embeddings from the fMRI data to see which movie was giving better results and generalization across subjects



Analysis

Opposite to that of encoding results, across all the video encoders, the movie **Lemonade** gave the best generalization and performance, followed by an almost equal performance by the movies, **iteration & defeat** with the movie **growth** giving the worst performance, consistently.

With and Without Normalization

Earlier we were running all the models by doing Z-Score normalisation across all the brain voxels and the video embeddings

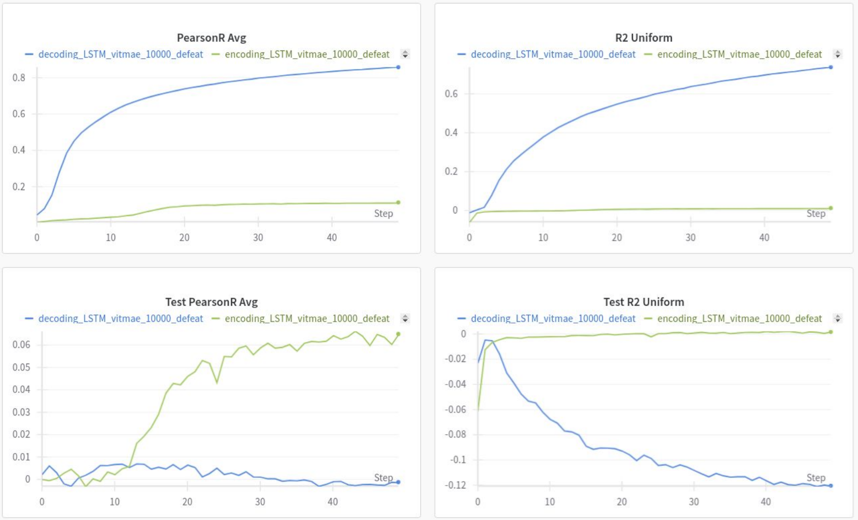
As asked by sir, we ran this experiment by removing the normalization for both of them

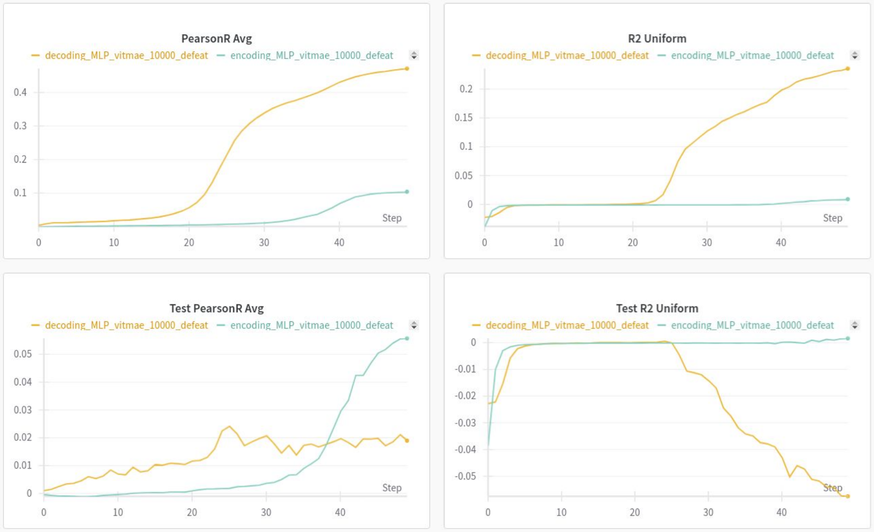
Analysis

- As we can see that, just by removing normalization while keeping everything else the exact same, we were able to see good generationalization for the movie **iteration** but a performance drop in the movie **defeat**.
- Therefore, unlike the encoding task, removing the normalization didn't help the decoding task as much.

Encoding vs Decoding

Encoding vs Decoding





Even, after multiple hyperparameter tunings, the decoding model was overfitting the train data, thus leading to a worse performance than the encoding models over all tasks, models, number of voxels

Classification

Classification

We tried two methods to classify the movie using the fMRI which was recorded while the participants were watching the movies and broke them into two experiments

- **Inter-subject** : We had independent training and testing sets having fMRI of different participants while they were watching the four movies and we tried to predict the movie using the test participants' fMRI data.
    - This lead to an accuracy which is slightly above the random chance level (of 25%) of 30%
- **Intra-subject :** We used 80% of the timestamps from all the participants into the train data and tried to predict the movie on the remaining 20% of the data for all the participants.
    - This lead to a whopping accuracy of 99.79%
    - *This was achieved with a 5 layer linear model*

Classification

Analysis

This makes sense as there was a lot of inter-subject variability leading to not much learning and generalization which could have been used for the inter-subject test

On the other hand, since the model had seen previous data from each participant, it had generalized well across all the participant data, thus leading it to perform much better when we saw the remaining part of their fMRI

Thank You.