4/30/2025

William Min

Capstone in Computing and Linguistics

Advised by Robert Frank

Decoding Human Syntactic Processing with Autoencoder-based Transformer Models

The high-dimensional nature of blood oxygen level dependent (BOLD) signals captured by functional magnetic resonance imaging (fMRI) raises an important question for researchers studying the brain—how to best compress such large and noisy signals into a more compact format while still preserving meaningful characteristics of the original data. The question is relevant both for practical and theoretical purposes. On the theoretical side, studying the structure of such a condensed representation might reveal patterns in brain activity, either over time or space. On the practical side, these representations are useful in building brain-computer interfaces because they offer an efficient way to provide such a model with meaningful information about brain activity.

fMRI signals tend to be quite large due to the fact that they represent 3 spatial dimensions across another time dimension. A typical 20-second fMRI time series might have around 10 scans, each composed of measurements from around 10^5 voxels. This means such a signal could be composed of around a million floating-point numbers. Considering that use of an fMRI machine for a single subject for a few hours can cost over $1000, the ratio of dimensions in the data format to the number of available data points will be quite large for most studies, leading to "curse of dimensionality" phenomena where statistical models have difficulty representing and making predictions from such datasets.

Autoencoder-generated embeddings show promise as a way to solve this problem. Using this method, a machine learning model called an autoencoder is trained to transform raw fRMI data into a smaller format (called an embedding) and is optimized to effectively reconstruct the original signal from these embeddings. Autoencoding techniques have the advantage of being able to map different subjects' fMRI data into the same embedding space, allowing them to generalize across different subjects, adapting to heterogeneity across brains. Competing methods

(such as principal component analysis and sparse dictionary learning), on the other hand, tend to struggle in this regard. These techniques try to understand brain activity by splitting it into the activation of many different fixed brain networks, but do not effectively account for how these networks can vary in spatial and temporal activity across subjects. (Zhao et al, 2023)

Zhao et al. (2023) introduce a temporally correlated autoencoder (TCAE) that seeks to improve autoencoder performance by using attention, a technique adapted from transformer language models, to incorporate insight from the time-linked aspect of fMRI data. They show that TCAE produced embeddings that are interpretable and sensitive to known functional anatomical networks. Furthermore, the embeddings can be used as effective input into a machine learning model trained for a brain state prediction task. On seven different categories of task, such as emotion, language, and working memory, Zhao et al. report training classifiers on TCAE embeddings that predict the category of task with accuracies as high as 86 percent.

In this paper, I adapt TCAE to the Narratives dataset, (Nastase et al, 2021) an fMRI dataset in which subjects listen to stories whose transcripts are timestamped and aligned with the fMRI data. With access to these transcripts, I develop a syntactic linguistic prediction task to investigate the extent to which the embeddings correspond to the syntactic level of linguistic structure. To complete this prediction task, I employ a two-part model. First, a TCAE is trained with access to fMRI sequences only (unsupervised learning). Then, a prediction model is built that translates from the TCAE's embeddings to the final predictions and trained with supervised learning. Critically, the supervised learning stage allows for the loss gradient to update the TCAE encoder module's weights. This allows for the resulting TCAE encoder weights to be compared with the weights generated from purely unsupervised learning (autoencoding). I expect to see that the supervised task optimization causes the TCAE encoders to be more attentive to parts of the brain related to that syntactic task in humans.

Although Zhao et al. analyze predictive tasks for ML models trained on TCAE encodings, their dataset (Human Connectome task fMRI) only allows them to predict between different classes of tasks, such as emotion, language, motor tasks. That dataset lacks the linguistic granularity to delve further into the correlation between neural activity and linguistic behavior. (Barch et al. 2013).

My goal in using a more fine-grained linguistic dataset is to show that TCAE embeddings are sensitive to functional brain networks and to identify networks that have important predictive significance on token-level linguistic tasks. By developing a methodology by which to associate linguistic tasks with functional networks, I hope to provide a way for researchers building predictive ML models of neural activity to verify that their models are attentive to the theoretically relevant neural regions.

Dataset

The narratives dataset (Nastase et al, 2021), contains fMRI scan sequences from English speakers as they listened to recorded English stories. The dataset contains data from 345 subjects and 27 stories, totaling around 43,000 words. For each story, the dataset provides the audio file and text transcript with both phoneme and word-level timestamps, making it possible to align fMRI scans with their related stimuli.

Data Preprocessing

I use the cleaned, smoothed, surface-projected fMRI data provided by Nastase et al. This corresponds to the directory /narratives/derivatives/afni-smooth/sub-xxx/func in their datalad dataset. For more details on fMRI preprocessing, see the methods section of Nastase et al.

For a given subject and story, I read this data using the nibabel python library (Brett et al, 2024) into an array of size (n, v) where n is the number of fMRI scans that were taken over the time course of that story and v is the number of vertices in the surface projection of a scan. I use the data provided in the fsaverage6 format, so v = 81,924. I then slice these arrays to produce sequences of 10 scans, arrays of size (10, 81924). This corresponds with a 15 second duration, as the fMRI's TR is 1.5 seconds. After slicing in this manner, I acquire 21,527 of these sequences across all stories and subjects.

Syntactic Prediction Task and Corresponding Brain Regions

Syntax, a level of linguistic structure that deals with how words and phrases combine to form sentences has been long thought to be especially related with regions in the left frontal cortex, such as the pars opercularis and pars triangularis, (Broca's area, collectively).

(Geschwind 1979). However, these regions likely also have functions related to other levels of linguistic structure.

Because syntactic subcategories are so important to building syntactic structure, my prediction task involves predicting sequences of words' syntactic subcategories (part of speech tags) from their corresponding fMRI embedding sequences. According to the lateralized theory of language competence promoted by Paul Broca, and supported by modern neuroimaging methods, most of a subject's syntactic processing should occur in these left frontal regions, thus a successful model should be especially sensitive to those regions to pick up on related BOLD-response patterns. (Broca, 1861), (Grodzinsky & Friederici, 2006). While I expect that a successful model's predictions would depend heavily on Broca's area, it's important to note that syntactic processing has been shown to take place in other areas as well, such as Wernicke's area and other parts of the superior temporal gyrus. (Grodzinsky & Friederici, 2006).

The labels for my supervised syntactic prediction tasks are part of speech tags obtained from the stories' transcript. These tags are generated in the Universal Dependencies Part of Speech (POS) tags format (de Marneffe et al, 2021) with the spaCy python library. (Honnibal et al, 2020). The Universal Dependencies format consists of seventeen unique tags each representing a different grammatical category. After converting words to tags, these tags are associated with their corresponding words in the time-aligned transcript. Tags associated with text that doesn't have a phonetic correlate, such as punctuation, are discarded. Some contracted words, such as "won't" or "haven't" are associated with an ordered sequence of more than one POS tag; these particular contracted words would each be represented by two tags: VERB and PART.

fMRI to Part of Speech Prediction Dataset

Datapoints for this predictive task have the following structure:

Input: An fMRI scan sequence for some subject listening to some story, from time t_s to t_e where $t_s - t_e = 15$ seconds.

Expected Output: The sequence of POS tags associated with words from that subject and story whose phonetic start times are after $t_s - z$ and whose end times are before t_e-z. z is a constant of

4.5 seconds that represents the hemodynamic delay between neural activity and the associated BOLD response captured by fMRI.

Note that the fMRI scans are taken at a constant interval of 1.5 seconds, whereas the words from which the POS tags are generated occur at the rate of natural speech. This means that there is not a one-to-one mapping between fMRI scans and words.

Assessing Prediction Performance

Per token perplexity will be my primary metric for how well a model predicts POS tags. Perplexity for a sequence of tokens of length N is defined as:

$$\prod_{i=1}^{n} p(x_i|x_{<i})^{-1/N}$$

Essentially, perplexity is the reciprocal of the geometric average of model-assigned probability for the correct token (given the previous tokens). A higher perplexity means that a model is assigning less probability to the correct tokens and is therefore performing poorly.

Because there are seventeen tags in the Universal Dependencies framework, and 3 extra tags for padding, sequence start, and sequence end, a model that guesses randomly should have a perplexity of 20.

Prediction Model Architecture

The predictive model begins with the sequence of input scans, which are fed into the encoder portion of the TCAE. The data is forward propagated up to the codes layer, where those codes are used as the cross-attention targets for the POS decoder (transformed into key and value vectors). The fMRI decoder part of the TCAE is not used in the prediction task.

The POS decoder module is made using a classic transformer decoder in the style of Vaswani et al, 2017. Given an unfinished output sequence of POS tags and a sequence of fMRI codes, it will give a probability distribution over next tags for each tag of the output sequence.

Conceptually, this architecture is similar to transformer-based machine translation approaches. It treats fMRI scan sequences like a foreign language, using the TCAE encoder to capture temporal dependencies and build up a representation that can be translated by the POS decoder into a target language, in this case, a sequence of POS tags. Practically, the most significant difference between the two tasks is that when translating between languages, the input and output sequences should represent roughly the same information. In this FMRI to POS task, however, the input fMRI sequence will contain vastly more information than the output POS sequence because it captures information from the entire brain, which necessarily processes much more than just linguistic information. So, the fMRI encoder must both capture temporal dependencies in the data (as a foreign language encoder would) and also filter out irrelevant information from nonlinguistic neural activity.

fMRI Autoencoder

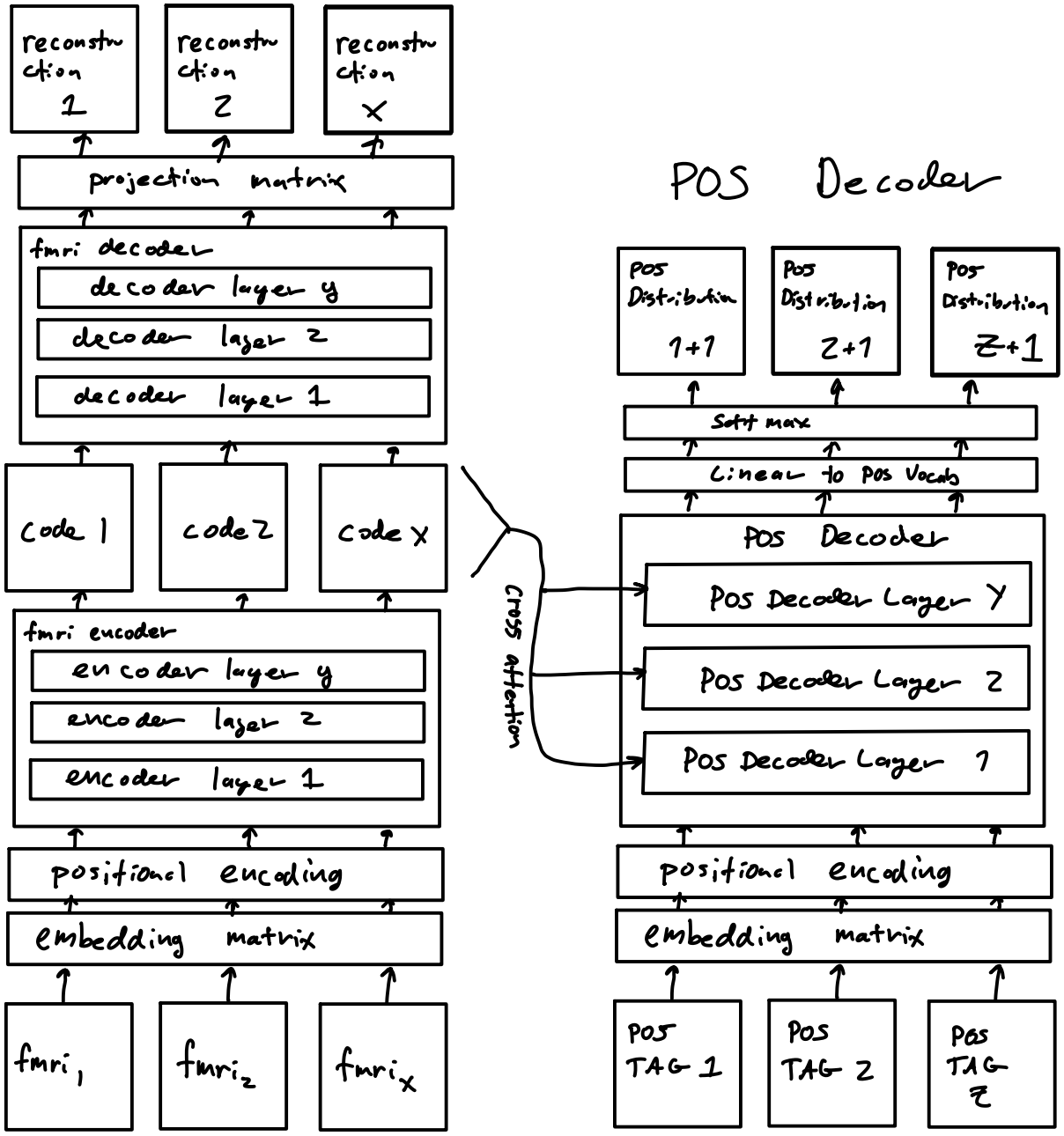


Fig 1 shows a diagram of the model architecture for the POS decoding task. On the left is the fMRI autoencoder (TCAE). On the right is the POS decoder.

Splitting

The Narratives dataset contains word sequences that are repeated because the same stories were tested on multiple subjects. This means that care must be taken when determining how to properly split the data for a language prediction task. If a particular sequence of words ends up in both the train, validation, and test splits of the dataset, then language model may just memorize the sequence in training, and if being tested with teacher forcing, recite the sequence once given enough true labels. Although conceptually, the corresponding datapoints in the train and test splits are unique because they have different associated BOLD activity, this would mean that the role of the brain data in the model could be as simple as a way for the model to identify which memorized POS sequence to recite. This would be undesirable because the model would not learn much about how syntax is represented in the brain.

To prevent such memorization of samples in my test set, I've decided to use a splitting technique that prevents sequences of words from being duplicated across training, validation, and test splits. I use random time splitting, as described in Xi et al, in which each POS/TR sequence for each story is randomly assigned to be in the train, validation, or test category. The data for all subjects for that sequence are then assigned to the same category. The data-leakage between temporally consecutive segments is not a major issue in this case because the sequences are long (15 seconds) and the POS labels are very different between consecutive segments.

TCAE Training

The training of the base fMRI autoencoder (TCAE) is roughly equivalent to the training process in Zhao et al, though the input data is of a different dimension (10, 81924).

A hyperparameter grid search was performed over learning rates, encoder & decoder layer counts, latent dimensions, head counts, and batch sizes. Validation loss curves for these hyperparameters appear in figure 2 below.

Training was performed on Yale's Grace computer cluster on two Nvidia a5000 GPUs.

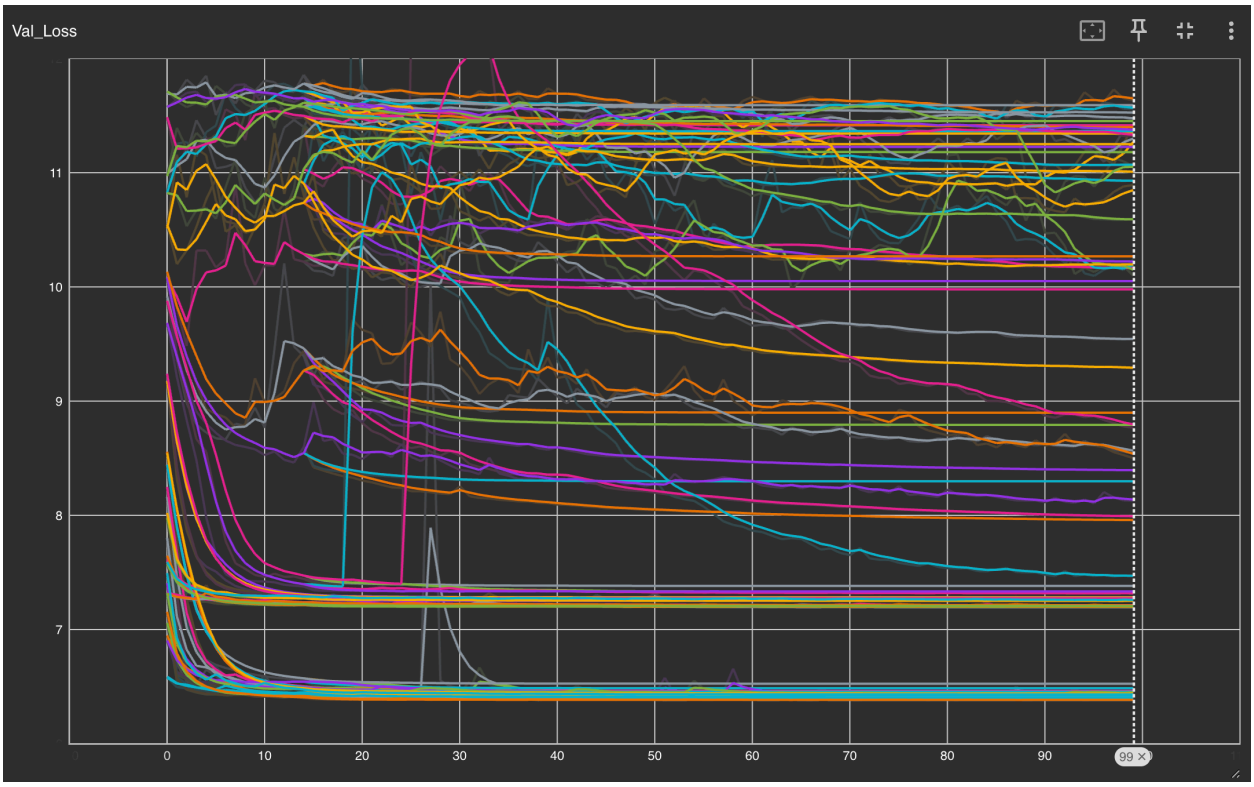


Fig 2 shows validation loss vs epochs for various hyperparameter combinations.

The best performing autoencoder had a learning rate of 1e-3, 4 transformer layers (2 for encoding, 2 for decoding), 128-digit latent dimension, and 8 attention heads. We see negligible improvements in loss for most models after 20 epochs of training.

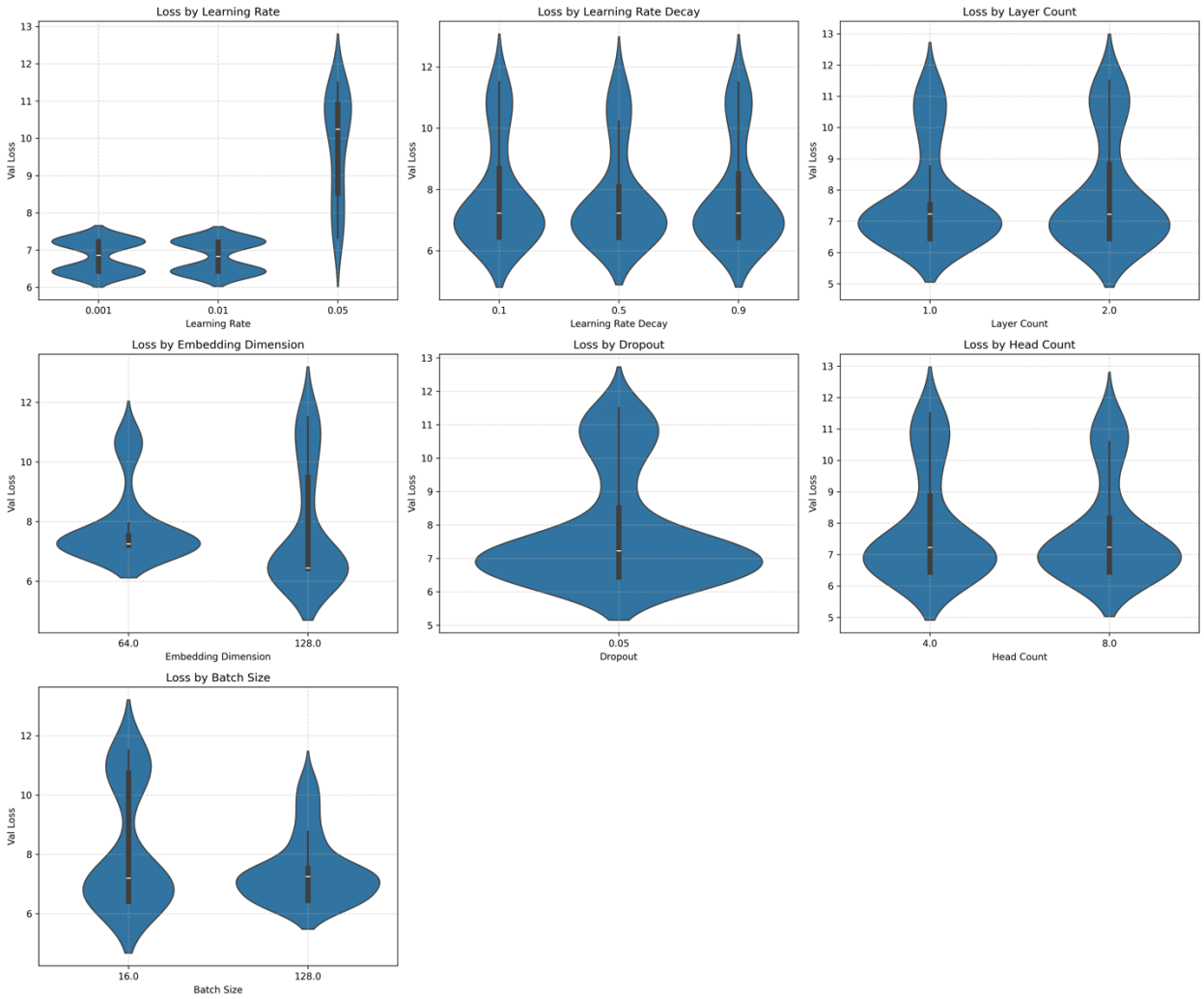


Fig 3. shows validation loss by grid step for the hyperparameters.

Note that model complexity did not seem to affect autoencoding performance very significantly. The distribution of loss was similar regardless of head count or layer count. However, embedding dimension size proved to be important with the 128-digit models tending to outperform the 64-digit ones. A learning rate of 0.05 proved too large for good autoencoding.

The 64-digit and 128-digit models with the lowest validation loss were selected as fMRI encoder “bases” to be coupled with a POS decoder. I refer to the 64 digit base as the “small” encoder and the 128 digit base as “large” encoder in future figures.

Prediction Model Freezing Strategies

When coupling the encoder portion of my fMRI autoencoder to my POS decoder, some subset of the weights in each model are selected to be untrainable or "frozen" throughout the predictor's training. This was done to test how the predictor's performance can improve or worsen when given more flexibility in how to process the brain and POS input.

For the fMRI autoencoder, the weights are set to be either all frozen, or all trainable. For the POS decoder, the weights are either set to be all trainable, or all frozen with the exception of the cross-attention weights. It would be undesirable to train the POS decoder with frozen cross-attention weights because it is intended to have the ability to make sense of the brain encodings. These two sets of two possibilities yield four freezing strategies for the entire model.

All models were trained for 50 epochs on Yale's research computing cluster with two Nvidia a5000 GPUs. For each freezing strategy, a hyperparameter grid search was performed over learning rates, learning rate decay, decoder layer counts, and fMRI encoder. The POS decoder was randomly initialized while the fMRI encoder was loaded from the previous autoencoding objective.

Overfitting

During the first training run, these predictor models severely overfit their training data. As models trained over epochs, they achieved good or even near-perfect performance on the training set, but their performance on the validation set continuously worsened. Reducing model complexity only helped flatten these curves—it did not allow models to improve their validation loss. This suggests that my data is too sparse on its own to train a model that performs well at POS prediction.

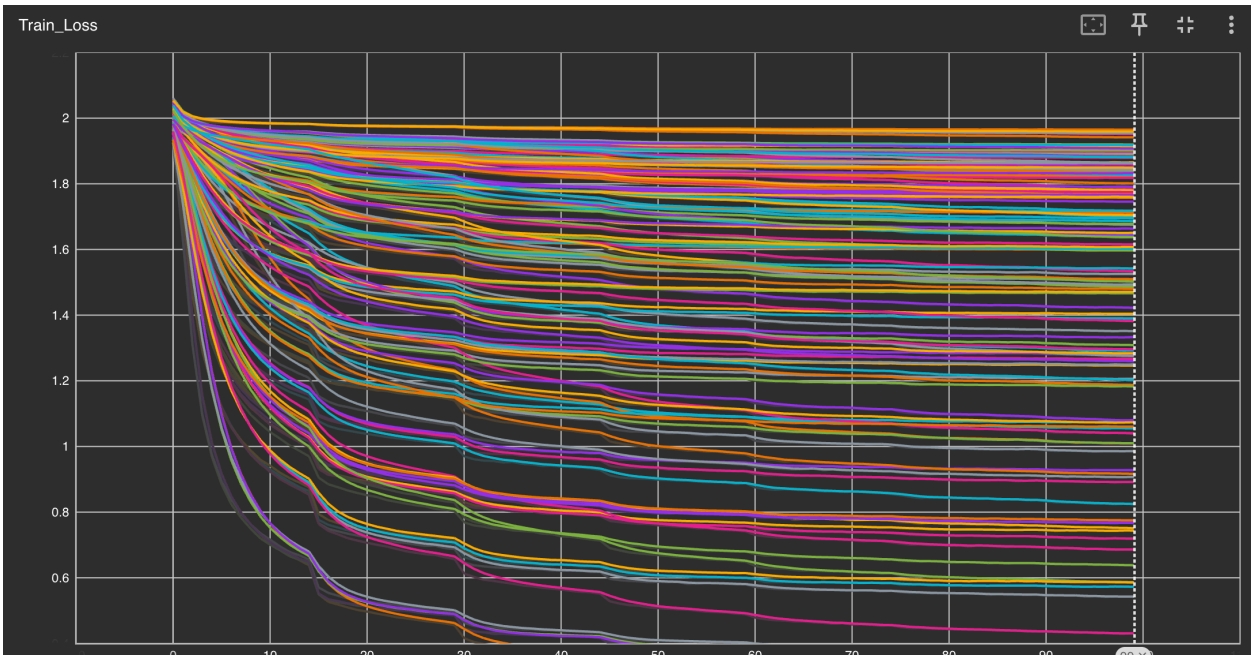


Fig 4. Train Loss vs epoch for various hyperparameter configurations and various freezing strategies

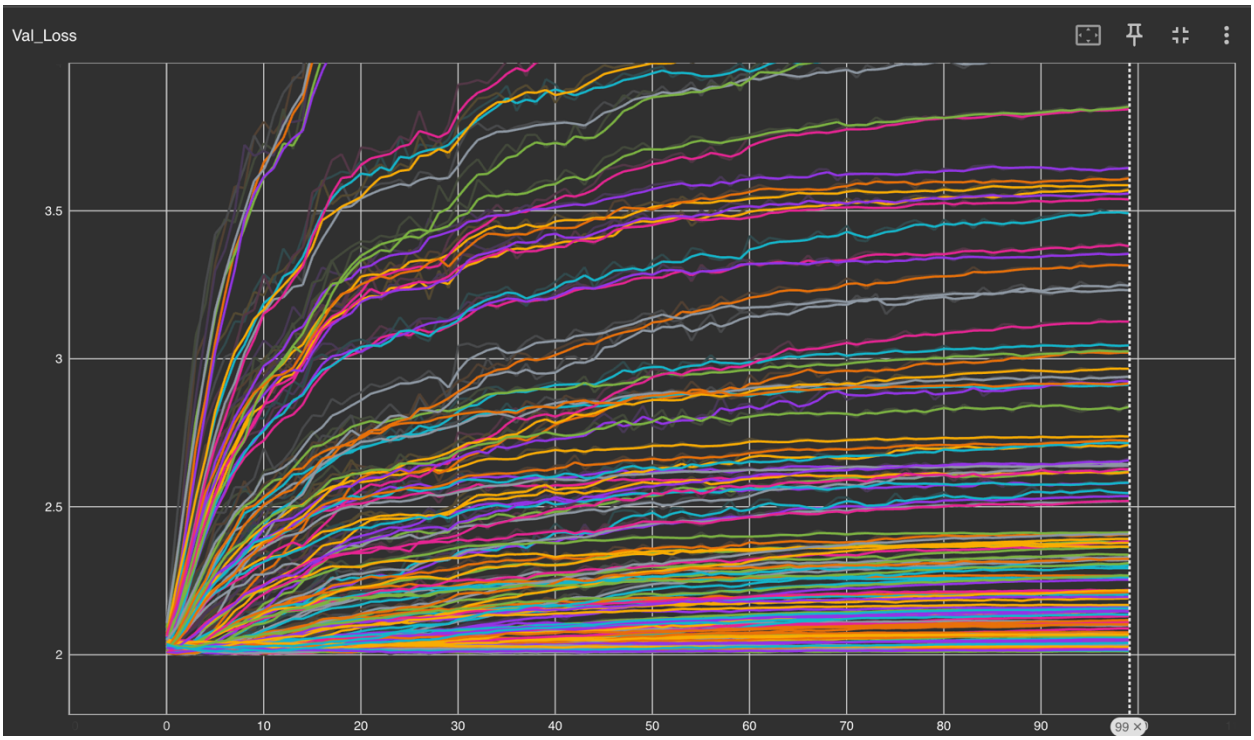


Fig 5. Validation loss vs epoch for various hyperparameter configurations and various freezing strategies

Language Pretraining

As a method to combat overfitting, I introduce a POS pretraining objective to my training pipeline. The goal of this objective is to pretrain my language decoder on a large external text dataset so that I can treat my POS-prediction task as a fine-tuning step. Considering how fine-tuning approaches have achieved success in large language modeling with relatively few datapoints, on the order of hundreds or thousands of examples, my dataset, consisting of some sixteen thousand datapoints seems reasonable. I leverage the external data as a way of remedying the sparseness of my dataset. If a model has some base knowledge about the syntactic structure language, it may be able to make better sense of limited examples of how syntax and BOLD activity are correlated. Essentially, I separate the tasks of learning how syntax works from that of learning how syntax relates to neural activity.

For this purpose, I use the wikitext-103-v1 dataset, which consists of nearly 2 million lines taken from English Wikipedia articles. (Merity et al, 2016). To convert the text to POS tokens, I again use spaCy. (Honnibal et al, 2020).

I use this text to train my POS decoder module, feeding tensors of zero into the cross-attention component. The effect of this is to skip the cross-attention module, relying entirely on the residual stream. Pretraining consisted of 1 epoch due to the large size of the training set. All decoder weights were set to be trainable.

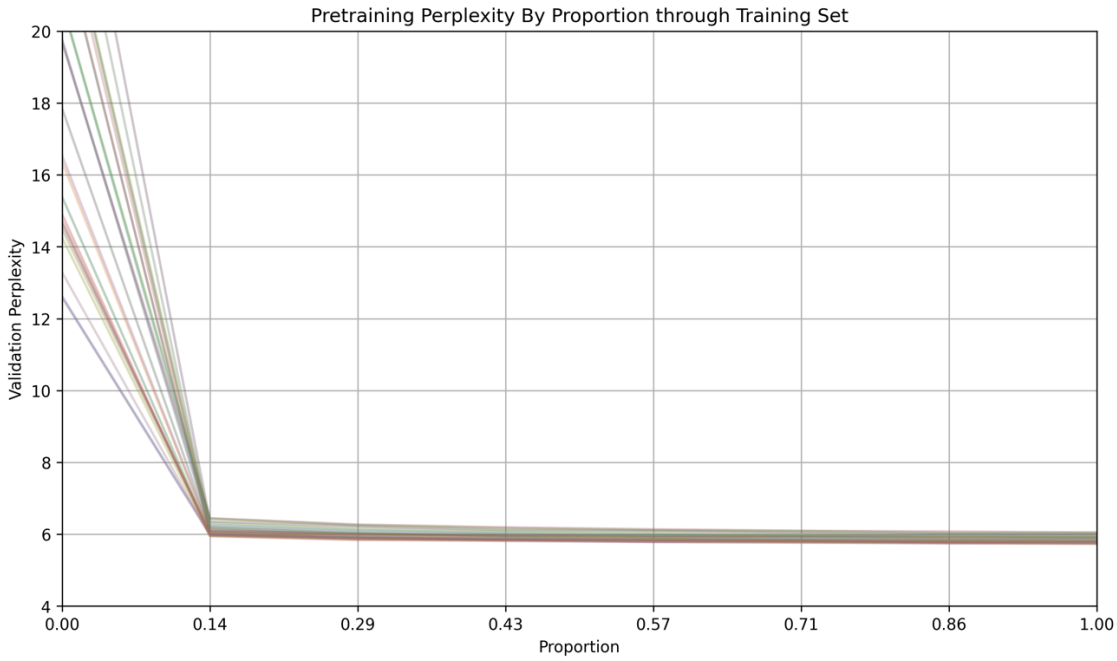


Fig 6. Pretraining Perplexity vs the proportion of the training set that had been used.

Combined Training

Once pretrained POS decoder bases were trained, they were coupled with fMRI encoders and trained on the POS-fMRI dataset for 50 epochs.

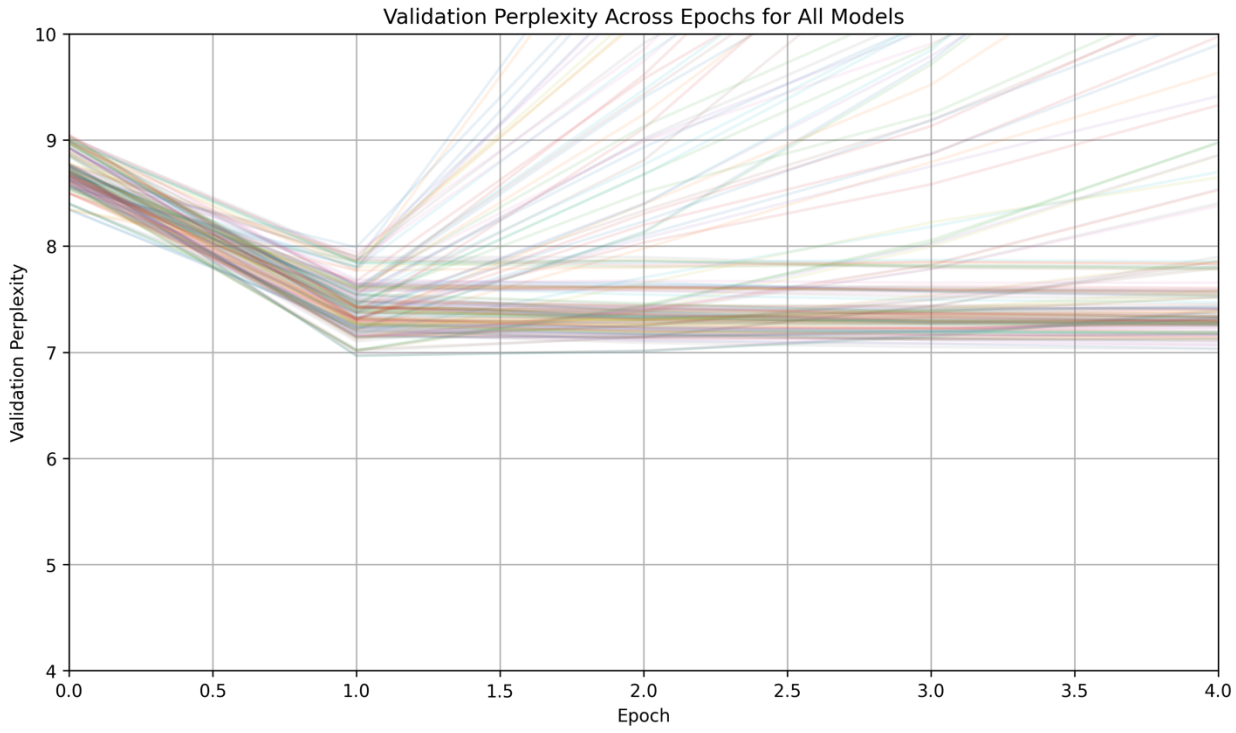


Fig 7. Validation Perplexity by epoch for predictor models. Nearly every model achieved its lowest perplexity after training on one epoch.

Note how the models' initial perplexities (in the high 8s range) on the Nastase dataset are higher than the final perplexities of the pretrained decoders. This can be attributed both to the facts that the models are receiving input to their cross-attention that they had not been subject to in pretraining, and that the pretraining text (snippets of Wikipedia articles) may be syntactically different from the text used in Nastase et al. (spoken stories).

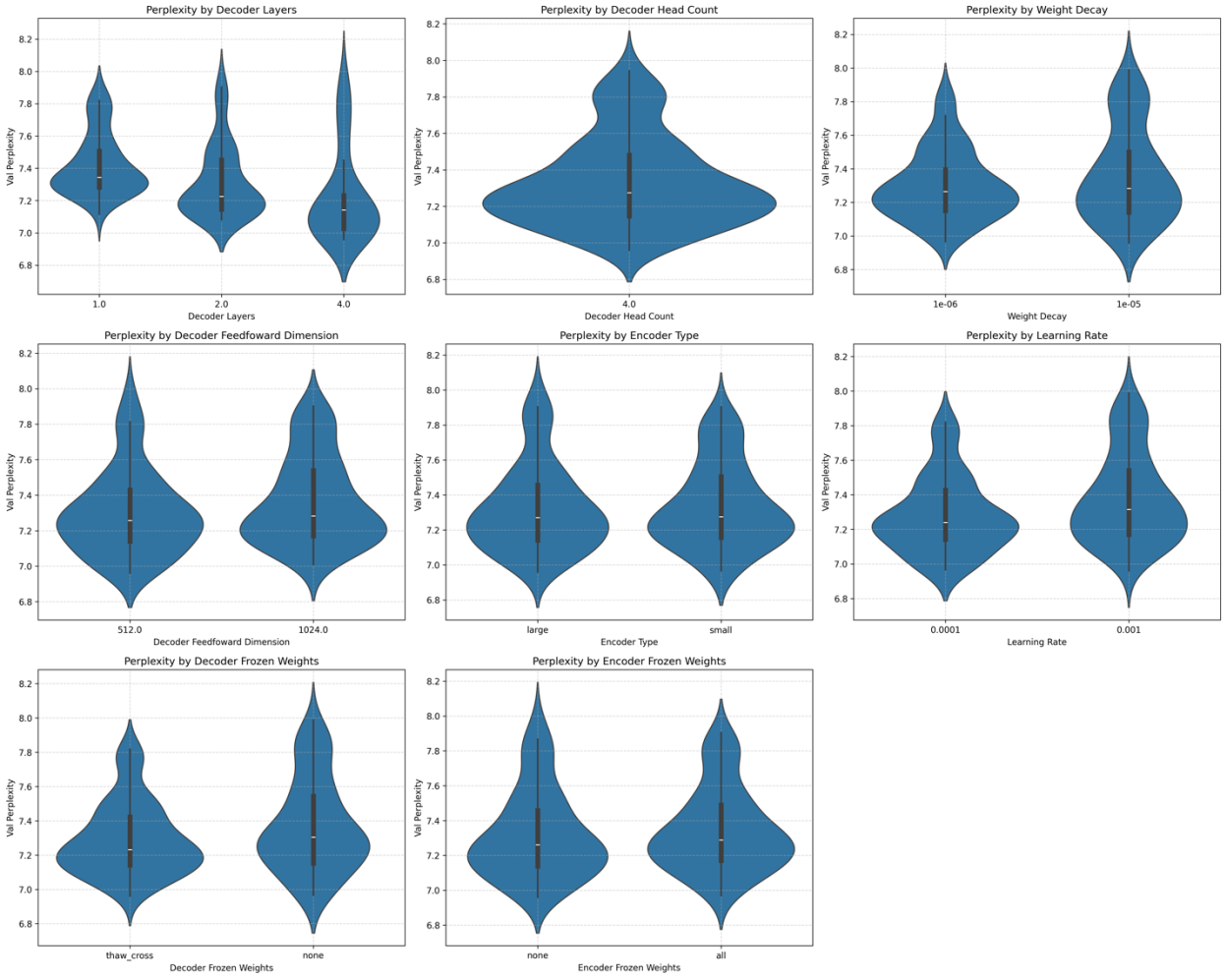


Fig 8. Predictor perplexity over hyperparameter values. "thaw_cross" indicates that only the cross_attention weights in the decoder are trainable.

A greater number of decoder layers appears to decrease perplexity, indicating that larger models perform better. Decoders where the non-cross-attention weights are frozen yield predictors that perform slightly better than those with decoders whose weights are all trainable. This suggests that limiting the decoder's ability to process its input can prevent it from overfitting the training examples.

On the contrary, among the best predictors, (perplexities 7 or lower), unfrozen encoders are more represented. This means that to achieve very good performance, encoders need the flexibility to tune their computation to better represent the input.

Permutation Testing

Predictor models improved their perplexity by around 1.5 perplexity points on average compared to the pretrained decoders. How much can this improvement be attributed to models learning how syntax is processed in the brain? Certainly, much of this improvement could just be due to models learning about the different kind of language in their new dataset and being able to ignore incoming fMRI encodings that they weren't used to in their pretraining.

To test this, I did a permutation test for each model. This consisted of randomly scrambling the associations between fMRI data sequences and POS sequences in the validation set. The result of this is a permuted validation set that maintains the same POS sequences and fMRI data sequences but breaks their associations. Then, I measure the difference between the perplexity on the original and permuted dataset. If the models have truly learned something about the association between syntax and the brain, then breaking this association should worsen their performance.

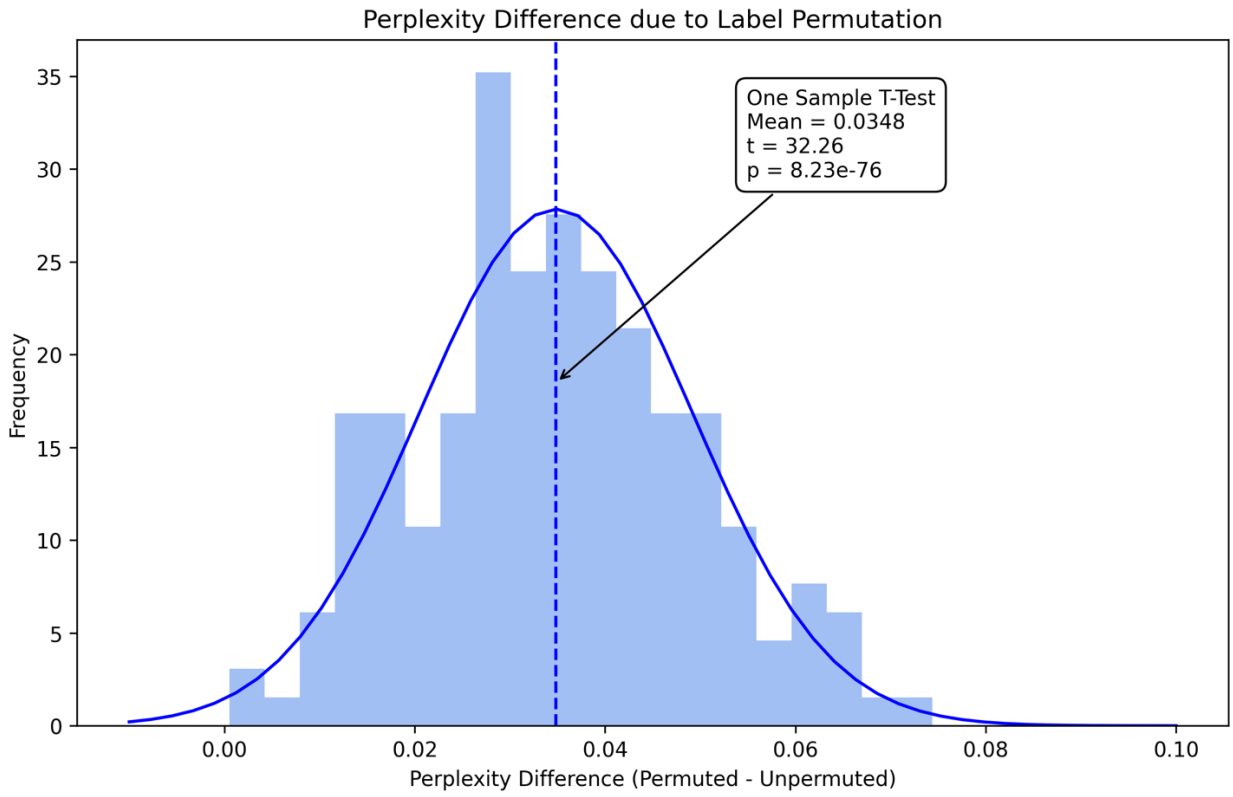


Fig 9 shows a histogram of perplexity differences.

The distribution of permutation test effects shown in figure 9 implies that the models do use the correlation between brain data and syntax to make their predictions, but that the size of this effect is very small. On average, permutation decreases their performance by only about a thirtieth of a perplexity point. The vast majority of the models' improvement from training on the brain data vs their baseline pretraining is due to learning about the structure of the syntax in the Nastase dataset, not the correlation between syntax and brain data.

Spatial Interpretability

Inspired by the methods used in Zhao et al, I interpret the models' fMRI encoder embedding matrices by examining how resulting embeddings depend upon activation in different regions of the brain. The embedding matrix for a model will be of size (81924, d_embedding). This means that one can interpret a value (i,j) in that matrix as being informative about how much digit j depends on vertex i. Because the magnitude of these values is more interpretable than the sign, I will square these values elementwise to get a measure of importance.

By looking at a column of this squared embedding matrix, I can see how a particular digit attends to the entire brain.

ROI relevance:

To compare these columns with anatomical regions of interest, I've downloaded the region maps from the Destrieux cortical atlas and converted them into Boolean vectors in the fsaverage6 format. (Destrieux, 2010) A 1 indicates that a vertex is in the region and a 0 indicates that it is not.

I define a measure of attentiveness that tracks how much a particular fMRI embedding matrix attends to a particular region of interest.

$$\text{mean\_attentiveness}(M,R) = \frac{1}{j}\sum_{c=1}^{j}(\text{cosine\_similarity}(M_{:,c}{}^2, R))$$

$$\text{where cosine\_similarity}(X,Y) = \frac{X^T Y}{\|X\|\|Y\|}$$

This score ranges from -1 to 1 with higher values indicating the model is more attentive to the region of interest. Mean attentiveness will represent the cosine similarity score for a squared weight column and a region averaged across embedding digits (column indices). The weight column is squared so that the mean attentiveness score recognizes both great positive and negative weights as representing more attentiveness to a vertex.

Results

For each region in the Destrieux atlas, I calculated the mean attentiveness for each model with a trainable encoder. I only use the models with the 128-digit embedding dimensions so that there is no effect from calculating means over different sizes of embedding dimension. Then, computing Pearson correlation between the trained models' perplexities and their mean attentiveness scores, I obtain a measure of how much that particular region affects prediction performance. Regions with a large negative correlation are of the most interest because for these regions, increased attentiveness leads to a decrease in perplexity, indicating that models may use these regions to obtain syntactic information to make better predictions.

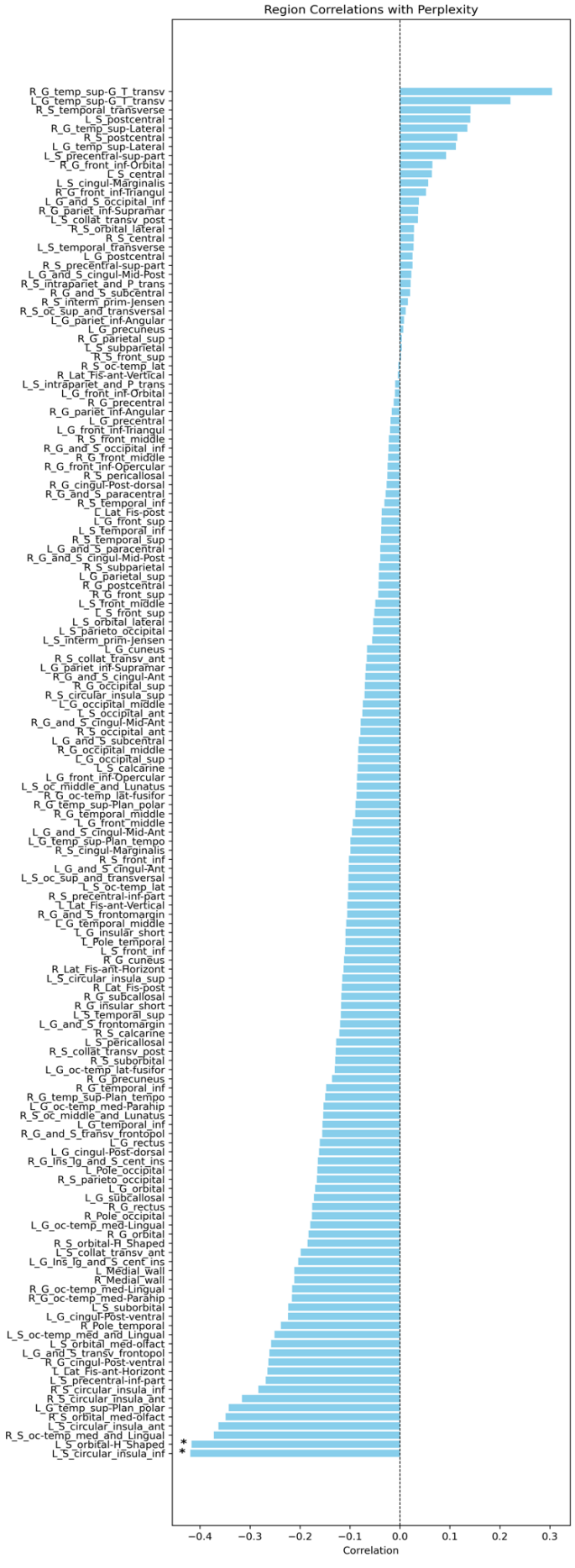

Region Correlations with Perplexity
R_G_temp_sup-G_T_transv
L_G_temp_sup-G_T_transv
R_S_temporal_transverse
L_S_postcentral
R_G_temp_sup-Lateral
R_S_postcentral
L_G_temp_sup-Lateral
L_S_precentral-sup-part
R_G_front_inf-Orbital
L_S_central
L_S_cingul-Marginalis
R_G_front_inf-Triangul
L_G_and_S_occipital_inf
R_G_pariet_inf-Supramar
L_S_collat_transv_post
R_S_orbital_lateral
R_S_central
L_S_temporal_transverse
L_G_postcentral
R_S_precentral-sup-part
L_G_and_S_cingul-Mid-Post
R_S_intrapariet_and_P_trans
R_G_and_S_subcentral
R_S_interm_prim-Jensen
R_S_oc_sup_and_transversal
L_G_pariet_inf-Angular
L_G_precuneus
R_G_parietal_sup
L_S_subparietal
R_S_front_sup
R_S_oc-temp_lat
R_Lat_Fis-ant-Vertical
L_S_intrapariet_and_P_trans
L_G_front_inf-Orbital
R_G_precentral
R_G_pariet_inf-Angular
L_G_precentral
L_G_front_inf-Triangul
R_S_front_middle
R_G_and_S_occipital_inf
R_G_front_middle
R_G_front_inf-Opercular
R_S_pericallosal
R_G_cingul-Post-dorsal
R_G_and_S_paracentral
R_S_temporal_inf
L_Lat_Fis-post
L_G_front_sup
L_S_temporal_inf
R_S_temporal_sup
L_G_and_S_paracentral
R_G_and_S_cingul-Mid-Post
R_S_subparietal
L_G_parietal_sup
R_G_postcentral
R_G_front_sup
L_S_front_middle
L_S_front_sup
L_S_orbital_lateral
L_S_parieto_occipital
L_S_interm_prim-Jensen
L_G_cuneus
R_S_collat_transv_ant
L_G_pariet_inf-Supramar
R_G_and_S_cingul-Ant
R_G_occipital_sup
R_S_circular_insula_sup
L_G_occipital_middle
L_S_occipital_ant
R_G_and_S_cingul-Mid-Ant
R_S_occipital_ant
L_G_and_S_subcentral
R_G_occipital_middle
L_G_occipital_sup
L_S_calcarine
L_G_front_inf-Opercular
L_S_oc_middle_and_Lunatus
R_G_oc-temp_lat-fusifor
R_G_temp_sup-Plan_polar
R_G_temporal_middle
L_G_front_middle
L_G_and_S_cingul-Mid-Ant
L_G_temp_sup-Plan_tempo
R_S_cingul-Marginalis
R_S_front_inf
L_G_and_S_cingul-Ant
L_S_oc_sup_and_transversal
L_S_oc-temp_lat
R_S_precentral-inf-part
L_Lat_Fis-ant-Vertical
R_G_and_S_frontomargin
L_G_temporal_middle
L_G_insular_short
L_Pole_temporal
L_S_front_inf
R_G_cuneus
R_Lat_Fis-ant-Horizont
L_S_circular_insula_sup
R_Lat_Fis-post
R_G_subcallosal
R_G_insular_short
L_S_temporal_sup
L_G_and_S_frontomargin
R_S_calcarine
L_S_pericallosal
R_S_collat_transv_post
R_S_suborbital
L_G_oc-temp_lat-fusifor
R_G_precuneus
R_G_temporal_inf
R_G_temp_sup-Plan_tempo
L_G_oc-temp_med-Parahip
R_S_oc_middle_and_Lunatus
L_G_temporal_inf
R_G_and_S_transv_frontopol
L_G_rectus
L_G_cingul-Post-dorsal
R_G_ins_lg_and_S_cent_ins
L_Pole_occipital
R_S_parieto_occipital
L_G_orbital
L_G_subcallosal
R_G_rectus
R_Pole_occipital
L_G_oc-temp_med-Lingual
R_G_orbital
R_S_orbital-H_Shaped
L_S_collat_transv_ant
L_G_ins_lg_and_S_cent_ins
L_Medial_wall
R_Medial_wall
R_G_oc-temp_med-Lingual
R_G_oc-temp_med-Parahip
L_S_suborbital
L_G_cingul-Post-ventral
R_Pole_temporal
L_S_oc-temp_med_and_Lingual
L_S_orbital_med-olfact
L_G_and_S_transv_frontopol
R_G_cingul-Post-ventral
L_Lat_Fis-ant-Horizont
L_S_precentral-inf-part
R_S_circular_insula_inf
R_S_circular_insula_ant
L_G_temp_sup-Plan_polar
R_S_orbital_med-olfact
L_S_circular_insula_ant
R_S_oc-temp_med_and_Lingual
L_S_orbital-H_Shaped
L_S_circular_insula_inf
*
*
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
Correlation

Figure 10 shows regions sorted by their mean attentiveness correlation with model perplexity. Negative correlation indicates that when a model's encoder attends more to a specific region, it tends to improve in performance. Stars indicate p-values less than 0.005. These values are **not** corrected for multiple comparisons.

While this method does not identify any regions whose correlations between attentiveness score and model perplexity is statistically significant when correcting for multiple comparisons, it still may highlight regions of interest that inform model predictions. Both regions that were significant before multiple comparison correction are located in the left hemisphere and have been shown to take part in linguistic processing.

The left inferior insular cortex, which contains the Destrieux region "L_S_circular_insul_inf" is directly above the arcuate fasciculus, a white-matter tract that connects Broca's and Wernicke's areas, facilitating language processing. Damage to this area is also linked with conduction aphasia. (Damaiso, 1980).

The left orbitofrontal cortex (lOFC), containing the Destrieux region "L_S_orbital-H_Shaped" has been shown through functional connectivity studies to be highly connected with known language-specific regions including Broca's area and the superior temporal gyrus. (Du J et al. 2020). Recent functional neuroimaging studies suggest that the lOFC underscores hemispheric language dominance and plays a "substantial role in human language functions." (Jiang et al. 2024).

Visualizing Regions of Interest

In addition to my correlation-based method, I use a method that compares the best performing models with the worst performing to visualize the vertices that are likely to be relevant to decoding syntactic processing.

My method is as follows: First, the best five and worst five models (with trainable encoders) are selected by their perplexities. Then, their embedding matrices are squared elementwise so that we obtain a measure of how extreme the weights are without regard to their sign. For each model, I then average these matrices across digits, giving us a map of mean

weight intensity by vertex. Then, I average within groups to produce two maps—one for the best models, and one for the worst models.

Finally, calculating the difference between these two maps (best minus worst), I produce a map that displays where increased weight intensity may improve model predictions.

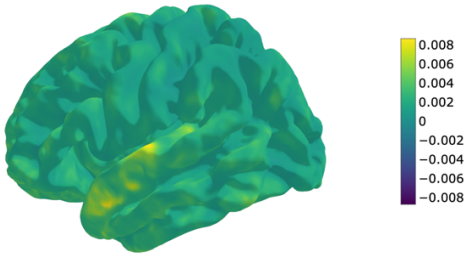
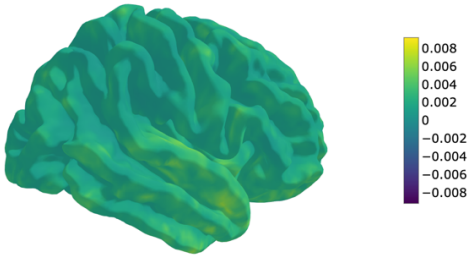
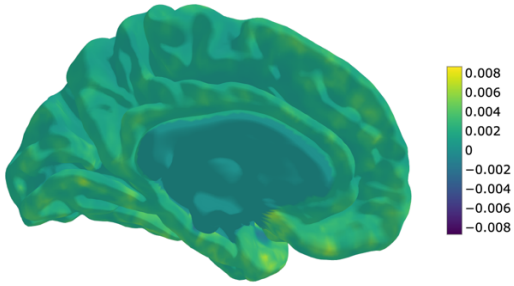
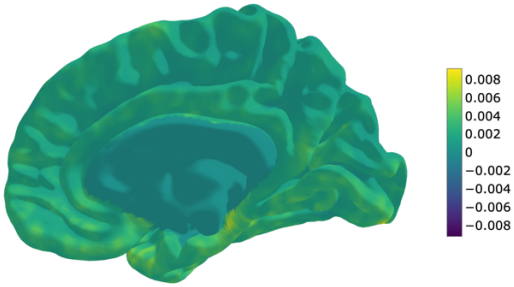
	Left Hemisphere	Right Hemisphere
Lateral View		
Medial View		

Figure 11 shows four views of a difference map between the best and worst predictor models. Positive (yellow) values indicate that more extreme weights at a given region may improve predictor performance. Negative (indigo) values indicate the opposite. The values by the colorbar represent mean squared weight values.

This difference map supports the hypothesis that these models respond to language lateralization in the brain—the left hemisphere shows swaths of positive values in the superior temporal gyrus (STG). While the right hemisphere shows some similar effect in the STG, the effect is less pronounced than in the left.

Furthermore, the map identifies the left STG as a region that likely plays an outsized role in informing model predictions. This corresponds to the widely accepted consensus that the STG plays a major role in syntactic processing. (Grodzinsky & Friederici, 2006).

Conclusion and Limitations

In this paper, I've shown that autoencoder-based transformer models can learn to use fMRI data from human participants to predict syntactic categories for words that those participants are hearing. By examining the embedding matrices used in the fMRI encoders of these models, I've shown that their performance is likely to depend on how attentive they are to certain language-related regions of the brain, such as the left inferior insular cortex, the orbitofrontal cortex, and the superior temporal gyrus.

I struggle to explain why these regions seem to play more of a role in predictor performance than other theoretically relevant regions such as Broca's area. My results are primarily qualitative and intended to highlight regions likely to influence model predictions. I leave it to future work to demonstrate a more rigorous statistical significance and to describe why these models' performance seems to be influenced more by some syntax-processing regions than others.

Code

Code is available at: https://github.com/minw1/TCAE-Narratives/tree/master.

Citations

Andersen, Anders, et al. Principal component analysis of the dynamic response measured by fMRI: a generalized linear systems framework, Magnetic Resonance Imaging, Volume 17, Issue 6, 1999,

Barch, Deanna M., et al. "Function in the Human Connectome: Task-fMRI and Individual Differences in Behavior." *NeuroImage*, vol. 80, 2013, pp. 169–189. Elsevier, https://doi.org/10.1016/j.neuroimage.2013.05.033.

Brett, M., et al. Nipy/nibabel: 5.3.1. 5.3.1, Zenodo, 15 Oct. 2024, doi:10.5281/zenodo.13936989.

Broca, P. (1861). Perte de la parole, ramouissement chronique et destruction partielle du lobe antérieur gauche du cerveau. Bulletins de la Société Anthropologique de Paris, 2, 235-238.

Défossez, A., Caucheteux, C., Rapin, J. *et al.* Decoding speech perception from non-invasive brain recordings. *Nat Mach Intell* **5**, 1097–1107 (2023). https://doi.org/10.1038/s42256-023-00714-5

Destrieux C, Fischl B, Dale A, Halgren E. Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. Neuroimage. 2010 Oct 15;53(1):1-15. doi: 10.1016/j.neuroimage.2010.06.010. Epub 2010 Jun 12. PMID: 20547229; PMCID: PMC2937159.

Dong, Q., Qiang, N., Lv, J., Li, X., Liu, T., Li, Q. (2020). Spatiotemporal Attention Autoencoder (STAAE) for ADHD Classification. In: Martel, A.L., *et al.* Medical Image Computing and Computer Assisted Intervention – MICCAI 2020. MICCAI 2020. Lecture Notes in Computer Science(), vol 12267. Springer, Cham. https://doi.org/10.1007/978-3-030-59728-3_50

Du J, Rolls ET, Cheng W, Li Y, Gong W, Qiu J, Feng J. Functional connectivity of the orbitofrontal cortex, anterior cingulate cortex, and inferior frontal gyrus in humans. Cortex. 2020 Feb;123:185-199. doi: 10.1016/j.cortex.2019.10.012. Epub 2019 Nov 16. PMID: 31869573.

Geschwind N. Specializations of the human brain. Sci Am. 1979 Sep;241(3):180-99. doi: 10.1038/scientificamerican0979-180. PMID: 493918.

Grodzinsky, Yosef and Friederici, Angela . Neuroimaging of syntax and syntactic processing, Current Opinion in Neurobiology,Volume 16, Issue 2,2006,Pages 240-246,ISSN 0959-4388, https://doi.org/10.1016/j.conb.2006.03.007.

Hanna Damasio, Antonio R. Damasio, The Anatomical Basis of Conduction Aphasia, *Brain*, Volume 103, Issue 2, June 1980, Pages 337–350, https://doi.org/10.1093/brain/103.2.337

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python. https://doi.org/10.5281/zenodo.1212303

Jiang X, Ma X, Sanford R, Li X. Adapting to Changes in Communication: The Orbitofrontal Cortex in Language and Speech Processing. Brain Sci. 2024 Mar 8;14(3):264. doi: 10.3390/brainsci14030264. PMID: 38539652; PMCID: PMC10969001.

Lu, Jason and Guo, Qingzhen. (2023) “The Double Helix inside the NLP Transformer.” https://arxiv.org/abs/2306.13817

Li, Qing, et al. "Simultaneous spatial-temporal decomposition of connectome-scale brain networks by deep sparse recurrent auto-encoders." *Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26*. Springer International Publishing, 2019.

Lin Zhao, Zihao Wu, Haixing Dai, Zhengliang Liu, Xintao Hu, Tuo Zhang, Dajiang Zhu, Tianming Liu, A generic framework for embedding human brain function with temporally correlated autoencoder, https://doi.org/10.1016/j.media.2023.102892.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, Daniel Zeman; Universal Dependencies. *Computational Linguistics* 2021; 47 (2): 255–308. doi: https://doi.org/10.1162/coli_a_00402

Nastase, S.A., Liu, YF., Hillman, H. *et al.* The “Narratives” fMRI dataset for evaluating models of naturalistic language comprehension. *Sci Data* **8**, 250 (2021). https://doi.org/10.1038/s41597-021-01033-3

Tong Y, Hocke LM, Frederick Bd. Short repetition time multiband echo-planar imaging with simultaneous pulse recording allows dynamic imaging of the cardiac pulsation signal. Magn Reson Med. 2014 Nov;72(5):1268-76. doi: 10.1002/mrm.25041. Epub 2013 Nov 22. PMID: 24272768; PMCID: PMC4198428.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. “Deep inside convolutional networks: Visualising image classification models and saliency maps.” arXiv preprint arXiv:1312.6034 (2013)

S. Merity, C. Xiong, J. Bradbury, R. Socher, Pointer sentinel mixture models, arXiv preprint arXiv:1609.07843 (2016).

Vaswani, A. "Attention is all you need." *Advances in Neural Information Processing Systems* (2017).

Xi, Nuwa et al. “UniCoRN: Unified Cognitive Signal ReconstructioN bridging cognitive signals and human language.” *Annual Meeting of the Association for Computational Linguistics* (2023).