

# **PUTTING IT ALL TOGETHER: INTERPRETING NEURAL NETWORK MODELS OF LANGUAGE PROCESSING IN THE HUMAN BRAIN**

Prof. Alexander Huth  
4.30.2020

# PROJECT PRESENTATIONS

- \* **Presentation schedule** is posted on canvas:
  - \* 6 presentations on Tuesday, May 5
  - \* 7 presentations on Thursday, May 7

# COURSE SURVEY

- \* If you have a couple minutes, please fill out the course instructor survey:  
<https://utdirect.utexas.edu/ctl/ecis/index.WBX>

# HOMEWORK 2 OFFICE HOURS

- \* Reminder that TA Mao Ye will be holding office hours **Friday, May 1, 10:30am - 12:00pm** to discuss homework 2 grading
  - \* Zoom link should be accessible on Canvas (under the “Zoom” tab)

# RECAP

- \* How can we understand the representations learned in (biological or) artificial neural networks?
  - \* **Feature visualization** aka *receptive field mapping*
    - \* *What types of inputs drive this unit?*
  - \* **Attribution**
    - \* *What is it about this input that drives this unit?*

# RECAP

- \* **Attribution** follows two strategies
  - \* Perturbation
  - \* Gradient attribution

# RECAP

- \* **Perturbation:**
  - \* Change something about the input. If the response of the selected unit changes, then the unit cared about the thing that was changed.
  - \* e.g. Bubbles

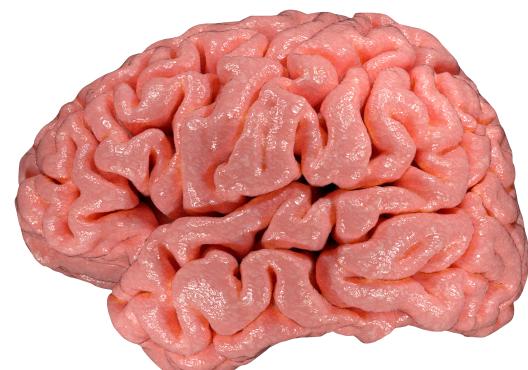
# RECAP

- \* **Gradient attribution:**
  - \* Compute the derivative of the selected unit activation with respect to each element (e.g. pixel) of the input. The magnitude of the derivative should be related to the importance of that input element.

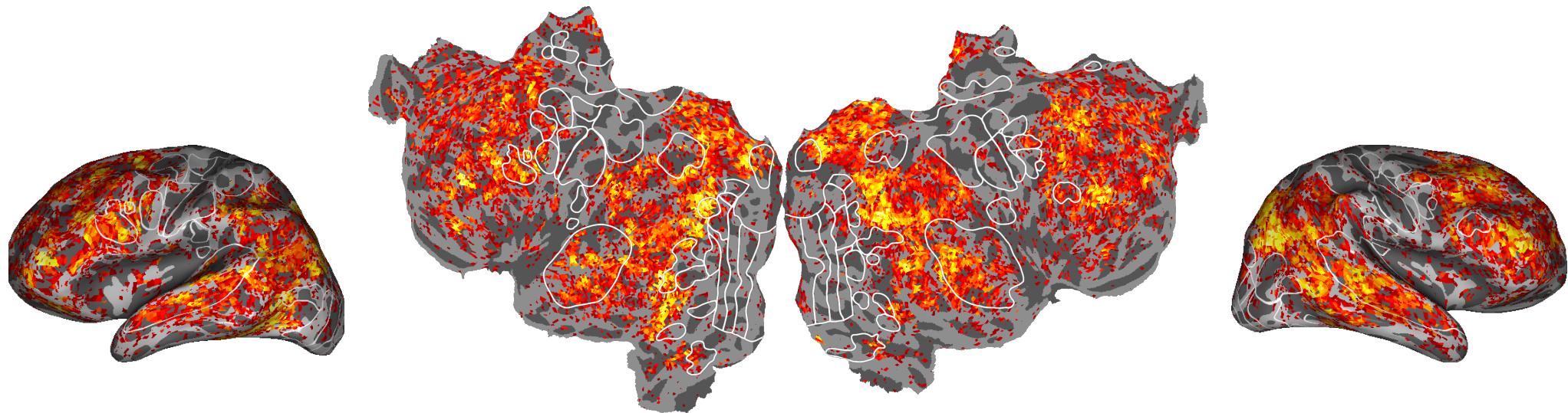
# LANGUAGE PROCESSING

- \* *How is the meaning of Language represented in the brain?*
- \* Let's do an experiment: record BOLD responses from subjects while they listen to natural language (stories)

*“Close your eyes and  
let the word paint a  
thousand pictures...”*



# LANGUAGE PROCESSING

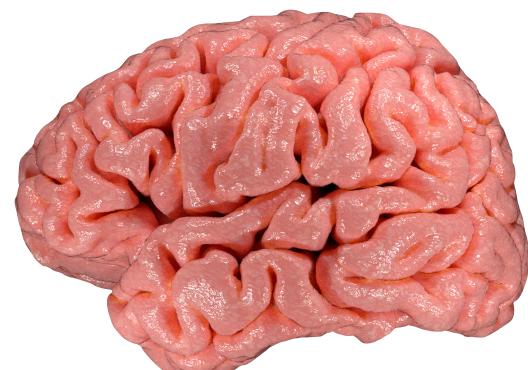


- \* Earlier studies (e.g. Huth et al., 2016): responses to natural language can be explained pretty well using **system identification models** based on **word embeddings**

# LANGUAGE PROCESSING

- \* Models based on word embeddings assume that the brain responds **independently** to each word
- \* How do we model **phrases**?

*“Word pictures let a  
and your thousand  
paint the eyes close...”*



# LANGUAGE MODELS

- \* One way to represent phrases is to use a **neural network language model**, which learns to predict the next word from the previous words
  - \* i.e. modeling  $P(w_t \mid w_{t-1}, w_{t-2}, w_{t-3}, \dots)$
  - \* *We talked about Long short-term memory (LSTM) Language models back in Lecture 19*

# LANGUAGE MODELS

- \* But the best language models use a different architecture: the **Transformer network** [*Vaswani et al., 2017*]
- \* Instead of recurrence, Transformers use a mechanism called **dot product self-attention**

# TRANSFORMERS

- \* Given: a sequence of words  $[w_1, \dots, w_t]$ , each represented as a  $d$ -dimensional embedding vector
- \* To predict word  $w_{t+1}$  we do the following:
  - \* First, multiply each word by 3 weight matrices:  $W_Q w_i = Q_i$ ,  $W_K w_i = K_i$ ,  $W_V w_i = V_i$
  - \* These are now called the **Query**, **Key**, and **Value**

# TRANSFORMERS

- \* Concatenate the vectors for all the words together into Query, Key, and Value matrices  $Q$ ,  $K$ , &  $V$  (each is  $d \times t$ )
- \* Use these matrices to compute **attention outputs**:

$$\text{Att}(Q, K, V) = \text{softmax}(QK^T)V$$

*attention weights,  
sum to 1 for each word*

# TRANSFORMERS

- \* This attention procedure is done many times (e.g. 12) in parallel with different  $W_Q$ ,  $W_K$ , &  $W_V$  matrices
  - \* Each one is called an *attention head*
- \* The outputs of all attention heads are concatenated and then multiplied by another weight matrix to get the layer output:

$$\text{Concat}(\text{Att}(Q^1, K^1, V^1), \text{Att}(Q^2, K^2, V^2), \dots)W_0$$

# TRANSFORMERS

- \* These transformer layers are then stacked: the output vector at each word replaces the word embedding, and the procedure is repeated
- \* Modern transformer networks are huge. The largest GPT-2 model has **48 layers** and **1.5B parameters**, and was trained on **~40GB of text** scraped from the internet

# TRANSFORMERS

- \* “But wait,” you might ask, “*how does this network know what order the words come in??!*”
  - \* There is *nothing* in this basic architecture that captures word order
- \* To give the network information about order, a **positional embedding vector** is added to the input embedding
  - \* One typical positional embedding uses sine and cosine functions with different frequencies:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

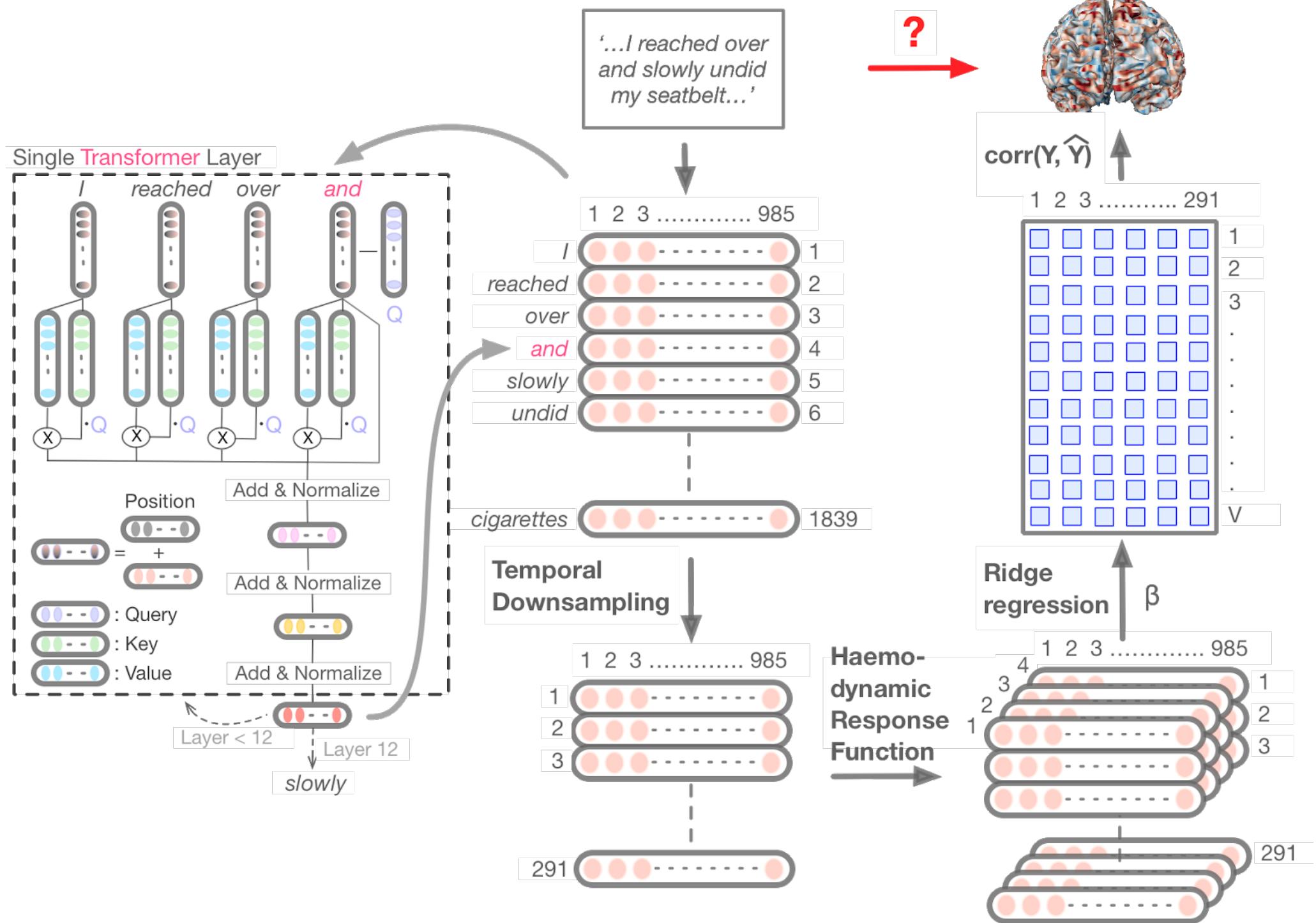
$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

# TRANSFORMERS

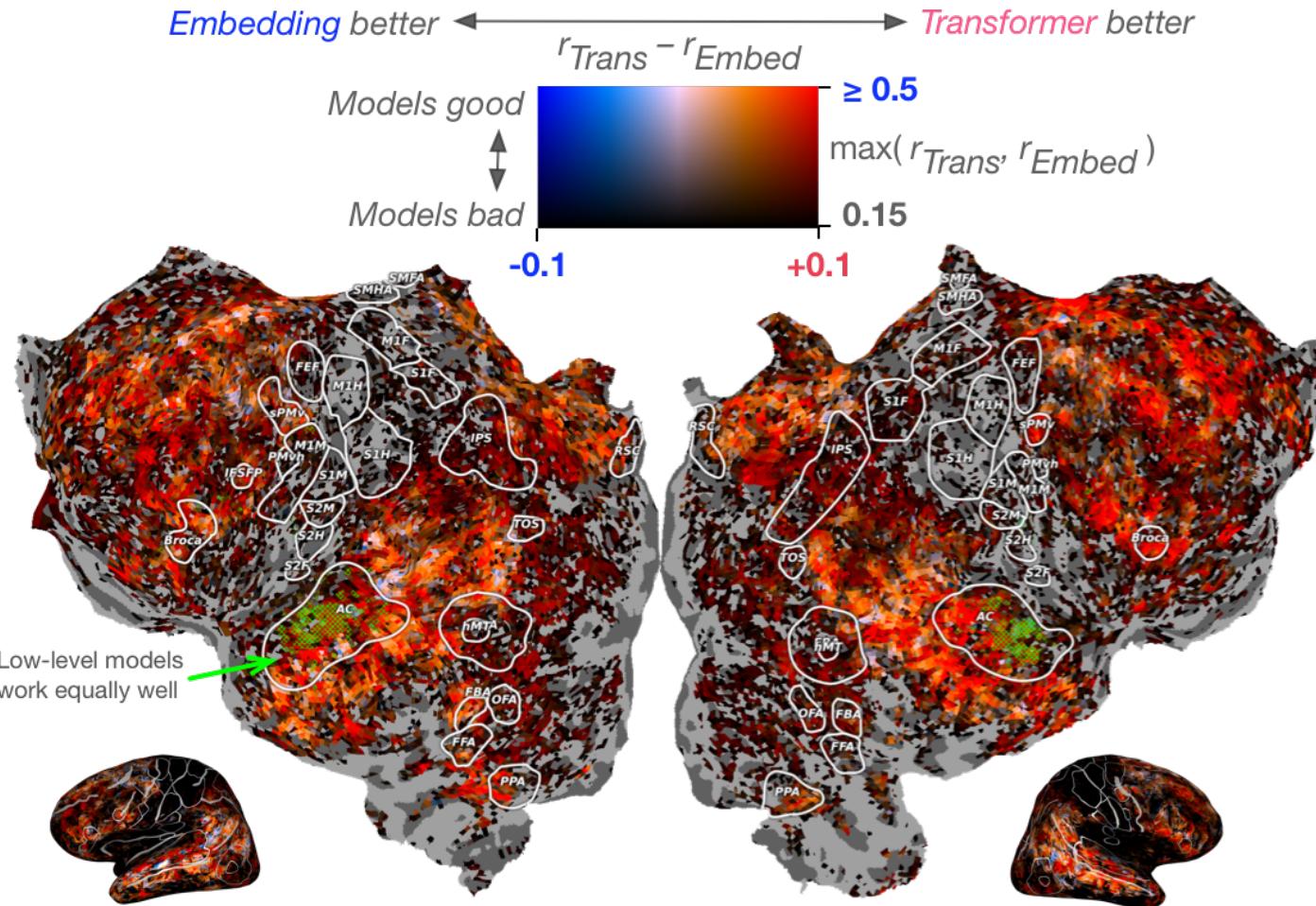
- \* Interesting aside: positional embeddings enable the transformer to act like a generalized convolutional network  
*[Cordonnier et al., 2020]*
- \* Depending on the learned weights, the network can accentuate information based on **relative position** (i.e. positional embedding) or based on **content**
- \* See <https://epfml.github.io/attention-cnn/>

# TRANSFORMER LANGUAGE MODEL: GPT

- \* To model how the brain responds to language, we (Shailee Jain) use a pretrained transformer language model called **GPT** [*Radford et al., 2018*]
  - \* GPT is a 12-layer transformer model with 12 attention heads per layer
- \* We can use these representations, instead of simple word embeddings, to predict brain activity



# TRANSFORMER LM →



- \* Transformer model substantially outperforms word embeddings for system identification

# TRANSFORMER LM →



- \* So how can we use this model to understand representations in the brain?
  - \* **Feature visualization:** which inputs make this “unit” active?
  - \* **Attribution:** what about this input makes this “unit” active?

# TRANSFORMER LM →



- \* Here a “unit” is a **voxel**, whose activation  $Y_i$  we model as the dot product between a set of weights  $\beta_i$  and a transformer representation  $T(X)$  of the phrase  $X$ :

$$Y_i \approx T(X)\beta_i$$

# TRANSFORMER LM →



- \* Let's do **feature visualization**:
  - \* **Build** library of phrases (140k)
  - \* **Predict** response to each ( $Y_i \approx \dots$ )
  - \* **Examine** phrases with highest predicted response

# TRANSFORMER LM →



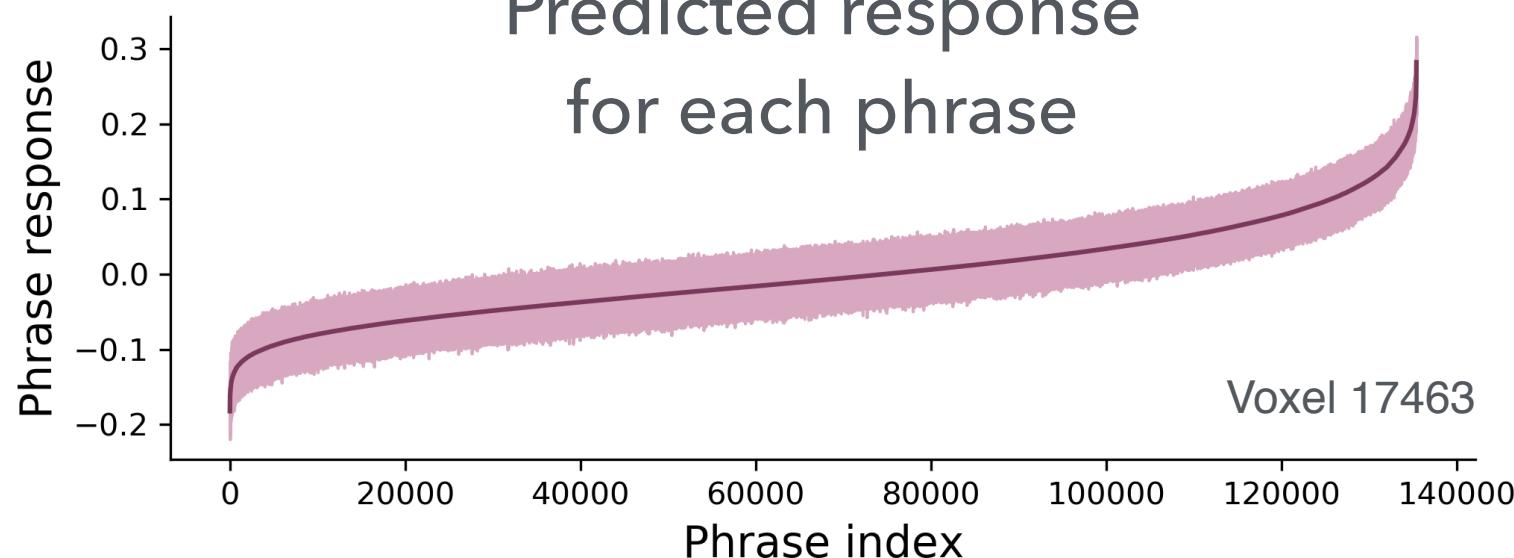
140,000 phrases  
(10 words each)



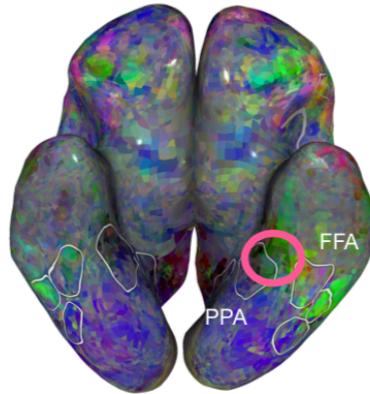
*Voxel  
Model*



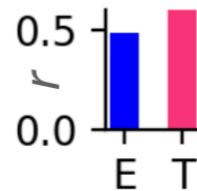
Predicted response  
for each phrase



# TRANSFORMER LM →



Voxel: 17463  
Parahippocampal



## Best Transformer Phrases

*and the only place to turn around  
on the bay bridge*

*pool with just grubby little  
changing rooms and a small cafe*

*got to Hardiwar our journey  
ended in a municipal car park*

*down the street and across the  
street at a coffee shop*

## Best Embedding Words

*shops*

*parking*

*building*

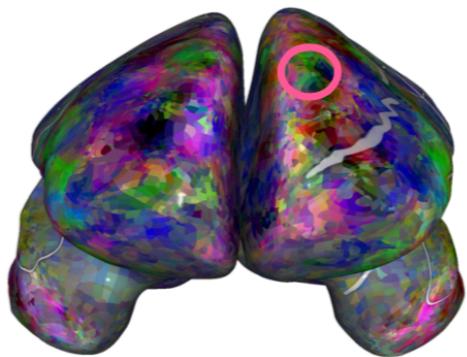
*campus*

*cafe*

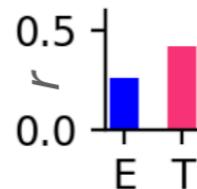
*street*

- \* This shows both **best phrases & best words** (acc. to embedding model). Both give ~similar understanding

# TRANSFORMER LM →



Voxel: 80162  
Superior frontal



## Best Transformer Phrases

*but I also knew that I wasn't  
doing well in law*

*a lesson in it I had been awarded  
a prestigious research*

*way these two guys are boy  
scouts they were literally eagle*

*artsy he seemed so exotic and  
interesting he was the creative*

## Best Embedding Words

*financial*

*working*

*cash*

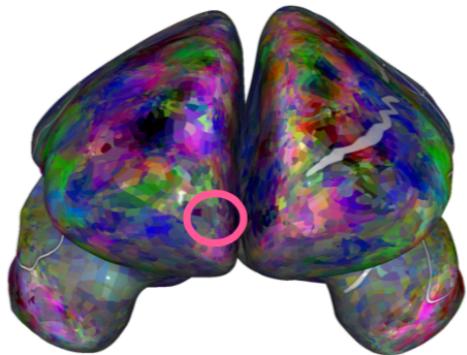
*employment*

*jobs*

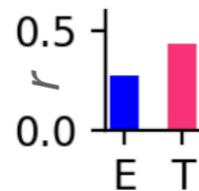
*investment*

- \* Here the best phrases are **more specific** than the best words

# TRANSFORMER LM →



Voxel: 42436  
Medial frontal



## Best Transformer Phrases

*motioned for me to sit next to him  
and thanked me*

*started to ask the lady a question  
she said excuse me*

*and she comes out and she says  
look I'm really sorry*

*you should probably go check in  
there so I say thanks*

## Best Embedding Words

*charming*

*gentleman*

*pleased*

*smiled*

*friendly*

*kindness*

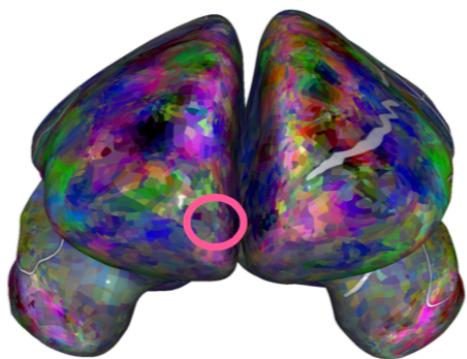
- \* Here the best phrases show **examples** while the best words **describe**

# TRANSFORMER LM →



- \* Let's do **attribution** using **perturbation**:
  - \* Take a 10-word phrase  $X$  and its predicted response  $Y_i(X)$
  - \* Generate 10 perturbed versions of the phrase  $[X'_1, \dots, X'_{10}]$ , each with one word replaced by a null word ('UNK')
  - \* Predict response to each perturbed phrase  $[Y_i(X'_1), \dots, Y_i(X'_{10})]$
  - \* Compute **importance** of word  $j$  in the phrase as  
$$\text{Importance}(X_j) = (Y_i(X) - Y_i(X'_{j})) / Y_i(X)$$

# TRANSFORMER LM →

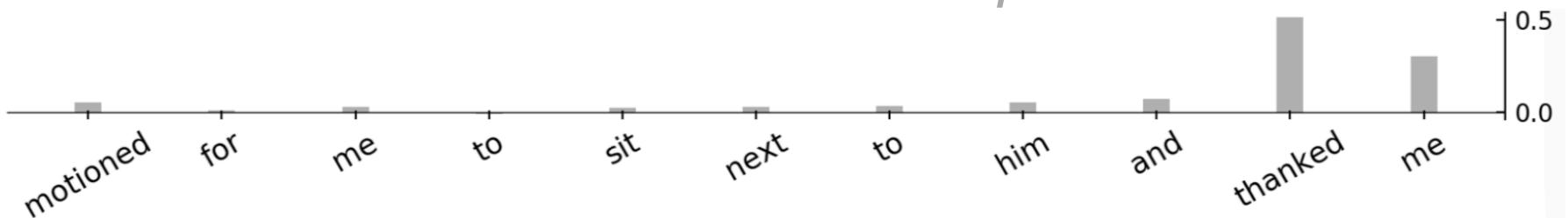


Voxel: 42436  
Medial frontal

Best Transformer Phrases

*motioned for me to sit next to him  
and thanked me*

*Importance score for  
each word in this phrase*



- \* This voxel's response hinges on the word “*thanked*”, suggesting **politeness** is important?

# TRANSFORMER LM →



- \* We combined **system construction**, **system identification**, **feature visualization**, and **attribution** techniques to help understand how the meaning of language is represented in the human brain
- \* What it all means? We're not sure yet, but we have the right tools to measure it!

# CONCLUSIONS

- \* In this class we've talked about:
  - \* **Neuroscience**, experimental methods
  - \* **Data quality** analyses
  - \* **System identification**
  - \* **Artificial neural networks** (convolutional, recurrent, LSTM, transformer)
  - \* **System construction**
  - \* **Interpretation methods** for ANNs (feature viz., attribution)

# CONCLUSIONS

- \* What I hope you've learned:
  - \* How ML methods (via system construction & identification) can be used to model computation in the brain
  - \* A bit about brains in general
  - \* A bit about ML methods in general

# CONCLUSIONS

- \* What I hope you take away:
  - \* An appreciation for **representations** as a key concept for understanding computation (both biological & artificial)
  - \* An appreciation of the difficulty of understanding & interpreting representations

# **NEXT TIME**

- \* Project presentations!