PNAS
www.pnas.org

Supplementary Information for

The neural architecture of language: Integrative modeling converges on predictive processing

Martin Schrimpf, Idan Blank, Greta Tuckute, Carina Kauf, Eghbal A. Hosseini, Nancy Kanwisher, Joshua Tenenbaum, Evelina Fedorenko

Martin Schrimpf <msch@mit.edu>, Evelina Fedorenko <evelina9@mit.edu>

This PDF file includes:

Supplementary text
Figures S1 to S11
Table S1

Code, data, models, and precomputed scores are available via www.github.com/mschrimpf/neural-nlp

Supplement

SI-Methods

1. Neural dataset 1: fMRI (*Pereira2018*). We used the data from Pereira et al.'s (2018) Experiments 2 (n=9) and 3 (n=6) (10 unique participants). (The set of participants is not identical to Pereira et al., 2018: i) one participant (tested at Princeton) was excluded from both experiments here to keep the fMRI scanner the same across participants; and ii) two participants who were excluded from Experiment 2 in Pereira et al., 2018, based on the decoding results in Experiment 1 of that study were included here, to err on the conservative side.) Stimuli for Experiment 2 consisted of 384 sentences (96 text passages, four sentences each), and stimuli for Experiment 3 consisted of 243 sentences (72 text passages, 3 or 4 sentences each). The two sets of materials were constructed independently, and each spanned a broad range of content areas. Sentences were 7-18 words long in Experiment 2, and 5-20 words long in Experiment 3. The sentences were presented on the screen one at a time for 4s (followed by 4s of fixation, with additional 4s of fixation at the end of each passage), and each participant read each sentence three times, across independent scanning sessions (see Pereira et al., 2018 for details of experimental procedure and data acquisition).

Preprocessing and response estimation: Data preprocessing was carried out with SPM5 (using default parameters, unless specified otherwise) and supporting, custom MATLAB scripts. (Note that SPM was only used for preprocessing and basic modeling—aspects that have not changed much in later versions; for several datasets, we have directly compared the outputs of data preprocessed and modeled in SPM5 vs. SPM12, and the outputs were nearly identical.) Preprocessing included motion correction (realignment to the mean image of the first functional run using 2nd-degree b-spline interpolation), normalization (estimated for the mean image using trilinear interpolation), resampling into 2mm isotropic voxels, smoothing with a 4mm FWHM Gaussian filter and high-pass filtering at 200s. A standard mass univariate analysis was performed in SPM5 whereby a general linear model (GLM) estimated the response to each sentence in each run. These effects were modeled with a boxcar function convolved with the canonical Hemodynamic Response Function (HRF). The model also included first-order temporal derivatives of these effects (which were not used in the analyses), as well as nuisance regressors representing entire experimental runs and offline-estimated motion parameters.

Functional localization: Data analyses were performed on fMRI BOLD signals extracted from the bilateral fronto-temporal language network. This network was defined functionally in each participant using a well-validated language localizer task (Fedorenko et al., 2010), where participants read sentences vs. lists of nonwords. This contrast targets brain areas that support 'high-level' linguistic processing, past the perceptual (auditory/visual) analysis. Brain regions that this localizer identifies are robust to modality of presentation (e.g., Fedorenko et al., 2010; Scott et al., 2017), as well as materials and task (Diachek et al., 2020). Further, these regions have been shown to exhibit strong sensitivity to both lexico-semantic processing (understanding individual word meanings) and combinatorial, syntactic/semantic processing (putting words together into phrases and sentences) [1]–[7]. Following prior work, we used group-constrained, participant-specific functional localization (Fedorenko et al., 2010). Namely, individual activation maps for the target contrast (here, sentences>nonwords) were combined with "constraints" in the form of spatial 'masks'—corresponding to data-driven, large areas within which most participants in a large, independent sample show activation for the same contrast. The masks (available from https://evlab.mit.edu/funcloc/ and used in many prior studies e.g., Jouravlev et al., 2019; Diachek et al., 2020; Shain et al., 2020) included six regions in each hemisphere: three in the frontal cortex (two in the inferior frontal gyrus, including its orbital portion: IFGorb, IFG; and one in the middle frontal gryus: MFG), two in the anterior and posterior temporal cortex (AntTemp and PostTemp), and one in the angular gyrus (AngG). Within each mask, we selected 10% of most localizer-responsive voxels (voxels with the highest *t*-value for the localizer contrast) following the standard approach in prior work. This approach allows to pool data from the same functional regions across participants even when these regions do not align well spatially. Functional localization has been shown to be more sensitive and to have higher functional resolution (Nieto-Castanon & Fedorenko, 2012) than the traditional group-averaging approach (Holmes & Friston, 1998), which assumes voxel-wise correspondence across participants. This is to be expected given the well-established inter-individual differences in the mapping of function to anatomy, especially pronounced in the association cortex (e.g., Frost & Goebel, 2012; Tahmasebi et al., 2012; Vazquez-Rodriguez et al., 2019).

We constructed a stimulus-response matrix for each of the two experiments by i) averaging the BOLD responses to each sentence in each experiment across the three repetitions, resulting in 1 data point per sentence per language-responsive voxel of each participant, selected as described above (13,553 voxels total across the 10 participants; 1,355 average, ±6 std. dev.), and ii) concatenating all sentences (384 in Experiment 2 and 243 in Experiment 3), yielding a 384x12,195 matrix for Experiment 2, and a 243x8,121 matrix for Experiment 3.

To examine differences in neural predictivity between the language network and other parts of the brain, we additionally extracted fMRI BOLD signals from two other networks: the multiple demand (MD) network (Duncan, 2010; Fedorenko et al., 2013) and the default mode network (DMN) (Buckner et al., 2008; Buckner & DiNicola, 2019). These networks were also defined functionally using well-validated localizer contrasts (Fedorenko et al., 2013; Mineroff et al., 2018) using a similar procedure as the one used for defining the language network: combining a set of 'masks' with individual activation maps, and selecting top 10% of most localizer-responsive voxels within each mask. Both networks were defined using a spatial working memory task (Fedorenko et al., 2011, 2013). For the MD network, we used the hard>easy contrast, and for the DMN network, we used the fixation>hard contrast. As for the language network, the MD and DMN masks were derived from large sets of participants for those contrasts, and are also available at https://evlab.mit.edu/funcloc/. The MD network and the DMN included 29,936 (2,994±230) and 10,978 (1,098±7) voxels, respectively.

2. Neural dataset 2: ECoG (*Fedorenko2016*). We used the data from Fedorenko et al.'s (2016) study (n=5). (The set of participants includes one participant, S2, who was excluded from the main analyses in Fedorenko et al., 2016 due to a small number of electrodes of interest; because we here used only language-responsiveness as the criterion for electrode selection, this participant had enough electrodes to be included.) Stimuli consisted of 80 hand-constructed 8-word long semantically and syntactically diverse sentences and 80 lists of nonwords (as well as some other stimuli not used in the current study). For the critical analyses, we selected a set of 52 sentences that were presented to all participants. The materials were presented visually one word at a time (for 450 or 700 ms), and participants performed a memory probe task after each stimulus (see Fedorenko et al., 2016 for details of the experimental procedure and data acquisition).

Preprocessing and response estimation: We here provide only a brief summary, highlighting points of deviation from Fedorenko et al. (2016). The total numbers of implanted electrodes were 120, 128, 112, 134, and 98 for the five participants, respectively. Signals were digitized at 1200 Hz. Similar to Fedorenko et al. (2016), i) the recordings were high-pass filtered with a cut off frequency of 0.5 Hz; ii) reference, ground, and electrodes with high noise levels were removed, leaving 117, 118, 92, 130, and 88 electrodes (for these analyses, we were more permissive with respect to noise levels compared to Fedorenko et al., 2016, to include as many electrodes in the analyses as possible; hence the numbers of analyzed electrodes are higher here than in the original study for 4 of the 5 participants); iii) spatially distributed noise common to all electrodes was removed using a common average reference spatial filter between electrodes with line noise smaller than a predefined threshold (electrodes connected to the same amplifier); and iv) a set of notch filters were used to remove the 60 Hz line noise and its harmonics. To extract the high gamma band activity—which has been shown to correspond to spiking neural activity in the vicinity of the electrodes [8]—we used a gaussian filter bank with centers at 73, 79.5, 87.8, 96.9, 107, 118.1, 130.4, and 144 Hz, and standard deviations of 4.68, 4.92, 5.17, 5.43, 5.7, 5.99, 6.3, and 6.62 Hz, respectively. This approach differs from Fedorenko et al. (2016), where an IIR band-pass filter was used to select frequencies in the range of 70-170 Hz, and is likely more sensitive (Dichter et al. 2018). Finally, as in Fedorenko et al. (2016), the Hilbert transform was used to extract the analytic signal [9] (except here, the average of the Hilbert signal across the eight filters was used as high-gamma signal), z-scored for each electrode with respect to the activity throughout the experiment, and the signal envelopes were downsampled to 300 Hz for further analysis (we did not additionally low-pass filter at 100 Hz, as in Fedorenko et al., 2016).

Functional localization: Mirroring the fMRI approach, where we focused on language-responsive voxels, data analyses were performed on signals extracted from language-responsive electrodes. These electrodes were defined in each participant using the same localizer contrast as in the fMRI datasets. In particular, we examined electrodes in which the envelope of the high gamma signal was significantly higher (at $p<.01$) for trials of the sentence condition than the nonword-list condition (for details, see Fedorenko et al., 2016).

We constructed a stimulus-response matrix by i) averaging the *z*-scored high-gamma signal over the full presentation window of each word in each sentence, resulting in 8 data points per sentence per language-responsive electrode (97 electrodes total across the 5 participants; 47, 8, 9, 15, and 18 for participants S1 through S5, respectively), and ii) concatenating all words in all sentences (416 words across the 52 sentences), yielding a 416x97 matrix.

To examine differences in neural predictivity between language-responsive and other electrodes, we additionally extracted high gamma signals from a set of 'stimulus-responsive' electrodes. Stimulus-responsive electrodes were defined as electrodes

in which the envelope of the high gamma signal for the sentence condition was significantly different (at $p<0.05$ by a paired-samples *t*-test) from the activity during the inter-trial fixation interval preceding the trial. This selection procedure resulted in 67, 35, 20, 29, and 26 electrodes. As expected, this set of electrodes included many of the language-responsive electrodes; for the analysis in SI Appendix SI-4, we exclude the language-responsive electrodes leaving 105 stimulus- (but not language-) responsive electrodes.

3. Neural dataset 3: fMRI (*Blank2014*). We used the data from Blank et al. (2014) (n=5). (The set of participants includes 5 of the 10 participants in Blank et al., 2014, because we wanted each participant to have been exposed to the same materials and as many stories as possible; the 5 participants included here all heard eight stories.) Stimuli consisted of stories from the publicly available Natural Stories Corpus (Futrell et al., 2018). These stories, adapted from existing texts (fairy tales and short stories) were designed to be "deceptively naturalistic": they contained an over-representation of rare words and syntactic constructions embedded in otherwise natural linguistic context. The stories were presented auditorily (each was ~5 min in duration), and following each story, participants answered 6 comprehension questions (see Blank et al., 2014 for details of the experimental procedure, data acquisition, and preprocessing).

Functional localization: As in the Pereira2018 dataset, data analyses were performed on fMRI BOLD signals extracted from the language network. From each language-responsive voxel of each participant, the BOLD time-series for each story was extracted. Across the eight stories, the BOLD time-series included 1,317 time-points (TRs, time of repetition; TR=2s and corresponds to the time it takes to acquire the full set of slices through the brain). To align the neuroimaging data with the story text, we first split the text into consecutive 2-second intervals (corresponding to the fMRI TRs) based on the auditory recording; if a word straddled boundaries of intervals, it was assigned to the 2s interval in which that spoken word ended. Each of the resulting intervals thus included a story "fragment", which could be a full short sentence, part of a longer sentence, or a transition between the end of one sentence and the beginning of another. Due to the temporal resolution of the HRF, whose peak's latency is 4-6 seconds, we assumed that each time-point in the BOLD signal represented activity elicited by the text fragment that occurred 4s (i.e., 2 TRs) earlier.

We constructed a stimulus-response matrix by i) averaging the BOLD signals corresponding to each TR in each story across the voxels within each ROI of each participant (averaging across the voxels within ROIs was done to increase the signal-to-noise ratio), resulting in 1 data point per TR per language-responsive ROI of each participant (60 ROIs total across the 5 participants), and ii) concatenating all story fragments (1,317 'stimuli'), yielding a 1,317x60 matrix.

4. Behavioral dataset: Self-paced reading (*Futrell2018*). We used the data from Futrell et al. (2018) (n=179). (The set of participants excludes 1 participant for whom data exclusions—see below—left only 6 data points or fewer.) Stimuli consisted of ten stories from the Natural Stories Corpus (same materials as those used in *Blank2014*, plus two additional stories), and any given participant read between 5 and all 10 stories. The stories were presented online (on Amazon's Mechanical Turk platform) visually in a dashed moving window display—a standard approach in behavioral psycholinguistic research [10]. In this approach, participants press a button to reveal each consecutive word of the sentence or story; as they press the button again, the word they just saw gets converted to dashes again, and the next word is uncovered. The time between button presses provides an estimate of overall language comprehension difficulty, and has been shown to be robustly sensitive to both lexical and syntactic features of the stimuli (Grodner & Gibson, 2005; Smith & Levy, 2013, inter alia) (see Futrell et al., 2018 for details of the experimental procedure and data acquisition.) We followed data exclusion criteria in Futrell et al. (2018): for any given participant, we only included data for stories where they answered 5 or all 6 comprehension questions correctly, and we excluded reading times (RTs) that were shorter than 100 ms or longer than 3000 ms.

We constructed a stimulus-response matrix by i) obtaining the RTs for each word in each story for each participant (848,762 RTs total across the 179 participants; 338 average, ±173 std. dev.), and ii) concatenating all words in all sentences (10,256 words across 485 sentences), yielding a 10,256x179 matrix.

5. Computational models. We tested 43 language models that were selected to sample a broad range of computational designs across three major types of architecture: embeddings, recurrent architectures, and attention-based 'transformer' architectures. Here we provide a brief overview (see SI Appendix Table SI-10 for a summary of key features varying across the

models). **GloVe** [11] is a word embedding model where embeddings are positioned based on co-occurrence in the Common Crawl corpus; **ETM** (Dieng et al., 2019, 20ng dataset) combines word embeddings with an embedding of each word's assigned topic; and **word2vec** [13]—abbreviated as w2v—provides embeddings which are trained to guess a word based on its context. **lm_1b** [14] is a 2-layer long short-term memory (LSTM) model trained to predict the next word in the One Billion Word Benchmark [15]; and the **skip-thoughts** model [16] is trained to reconstruct surrounding sentences in a passage. For all 38 transformer models (pretrained models from the HuggingFace library [17]), we only evaluate the encoder and not the decoder; the encoders process long contexts (100s of words) with a deep neural network stack of multiple attention heads that operate in a feed-forward manner (except the Transformer-XL-wt103 and the two XLNet models, which use recurrent processing), and differ mostly in the choice of directionality, network architecture, and training corpora (SI Appendix Table SI-11). We highlight key features of different classes of transformer models (BERT, RoBERTa, XLM, XLM-RoBERTa, Transformer-XL-wt103, XLNet, CTRL, T5, AlBERT, and GPT) in the order in which they appear in the bar-plots (e.g., Fig. 2a), except for the three 'distilled' models [18], which we mention in the end. **BERT** transformers [19] (n=4; bert-base-uncased, bert-base-multilingual-cased, bert-large-uncased, bert-large-uncased-whole-word-masking) are optimized to train bidirectional representations taking into account context both to the left and right of a masked token. **RoBERTa** transformers [20] (n=2; roberta-base, roberta-large) as a variation of BERT improve training hyper-parameters such as masking tokens dynamically instead of always masking the same token. **XLM** models [21] (n=7; xlm-mlm-enfr-1024, xlm-clm-enfr-1024, xlm-mlm-xnli15-1024, xlm-mlm-100-1280 , xlm-mlm-en2048) learn cross-lingual models by predicting the next ("clm") or a masked ("mlm") token in a different language. **XLM-RoBERTa** [22] (n=2; xlm-roberta-base, xlm-roberta-large) combines RoBERTa masking with cross-lingual training in XLM. **Transformer-XL-wt103** [23] adds a recurrence mechanism to GPT (see below) and trains on the smaller WikiText-103 corpus. **XLNet** transformers [24] (n=2; xlnet-base-cased, xlnet-large-cased) permute tokens in a sentence to predict the next token. **CTRL** [25] adds control codes to GPT (see below) which influence text generation in a specific style. **T5** transformers [26] (n=5; t5-small, t5-base, t5-large, t5-3b, t5-11b) train the same model across a range of tasks including the prediction of multiple corrupted tokens, GLUE [27], and SuperGLUE [28] in a text-to-text manner where the task is provided as a text prefix. **AlBERT** transformers [29] (n=8; albert-base-v1, albert-large-v1, albert-xlarge-v1, albert-xxlarge-v1, albert-base-v2, albert-large-v2, albert-xlarge-v2, albert-xxlarge-v2) use parameter-sharing and model inter-sentence coherence. **GPT** transformers (n=5) are trained to predict the next token in a large dataset emphasizing document quality (openaigpt [30] on the Book Corpus dataset, gpt2, gpt2-medium, gpt2-large, and gpt2-xl [31] on WebText). Finally, **distilled versions** of models [18] (n=3; distilbert-base-uncased, distilgpt2, distilroberta-base) train compressed models on a larger teacher network.

To retrieve model representations, we treated each model as an experimental participant (Figure 1) and ran the same experiment on it that was run on humans. Specifically, sentences were fed in sequentially into the model (for Pereira2018, Blank2014, and Futrell2018, sentences were grouped by topic / story to approximate the procedure with human participants). For embedding and recurrent models, sentences were fed in word-by-word; for transformers, the context before (but not after) each word was also fed into the models due to their lack of memory; the length of the context was determined by the models' architectures. For recurrent models, the memory was reset after each story (*Pereira2018, Blank2014* and *Futrell2018*), or each sentence (*Fedorenko2016*).

After the processing of each word, we retrieved ("recorded") model representations at every computational block (e.g., one LSTM cell or one Transformer encoder block). (Word-by-word processing increases computational cost but is necessary to avoid bidirectional models, like the BERT transformers, seeing the future.) When comparing against human recordings spanning more than one word such as a sentence (*Pereira2018*) or story fragment (*Blank2014*), we aggregated model representations: for the embedding models, we used the mean of the word representations; for recurrent and transformer models, we used the representation of the last word since these models already aggregate representations of the preceding context, up to a maximum context length of 512 tokens.

6. Comparison of models to brain measurements. We treated the model representation at each layer separately and tested how well it could predict human recordings (for *Pereira2018*, we treated the two experiments separately, but averaged the results across experiments for all plots except Fig. 2c). To generate predictions, we used 80% of the stimuli (sentences in *Pereira2018*, words in *Fedorenko2016* and *Futrell2018*, and story fragments in *Blank2014*; Fig. 1) to fit a linear regression from the corresponding 80% of model representations to the corresponding 80% of human recordings. We applied the

regression on model representations of the held-out 20% of stimuli to generate model predictions, which we then compared against the held-out 20% of human recordings with a Pearson correlation. This process was repeated five times, leaving out different 20% of stimuli each time, and we computed the per-voxel/electrode/ROI mean predictivity across those five splits. We aggregated these per-voxel/electrode/ROI scores by taking the median of scores for each participant's voxels/electrodes/ROIs and then computing the median and median absolute deviation (m.a.d.) across participants (over per-participant scores). Finally, this score was divided by the estimated ceiling value (see Estimation of ceiling below) to yield a final score in the range of $[0, 1]$. We report the results for the best-performing layer for each model (SI Appendix SI-12) but controlled for the generality of layer choices in train/test splits (SI Appendix Fig. S2b,c).

7. Estimation of ceiling. Due to intrinsic noise in biological measurements, we estimated a ceiling value to reflect how well the best possible model of an average human could perform. To do so, we first subsampled—for each dataset separately—the data with n recorded participants into all possible combinations of s participants for all $s \in [2, n]$ (e.g. {2, 3, 4, 5} for *Fedorenko2016* with n=5 participants). For each subsample s, we then designated a random participant as the target that we attempt to predict from the remaining $s - 1$ participants (e.g., predict 1 subject from 1 (other) subject, 1 from 2 subjects, …, 1 from 4, to obtain a mean score for each voxel/electrode/ROI in that subsample. To extrapolate to infinitely many humans and thus to obtain the highest possible (most conservative) estimate, we fit the equation $v = v_0 \times \left(1 - e^{-\frac{x}{\tau_0}}\right)$ where x is each subsample's number of participants, v is each subsample's correlation score and v_0 and τ_0 are the fitted parameters for asymptote and slope respectively. This fitting was performed for each voxel/electrode/ROI independently with 100 bootstraps each to estimate the variance where each bootstrap draws x and v with replacement. The final ceiling value was the median of the per-voxel/electrode/ROI ceilings v_0.

For *Fedorenko2016*, a ceiling was estimated for each electrode in each participant, so each electrode's raw value was divided by its own ceiling value. Similarly, for *Blank2014*, a ceiling was estimated for each ROI in each participant, so each ROI's raw value was divided by its own ceiling value. For *Pereira2018*, we treated the two experiments separately, focusing on the 5 participants that completed both experiments to obtain full overlap in the materials for each participant, and used 10 random sub-samples to keep the computational cost manageable. A ceiling was estimated for all voxels in the 5 participants who participated in both experiments. Each voxel's raw predictivity value was divided by the average ceiling estimate (across all the voxels for which it was estimated). For *Futrell2018*, given the large number of participants and because most participants only had measurements for a subset of the stimuli, we did not hold out one participant but rather tested how well the mean RTs for one half of the participants predicted the RTs for the other half of participants. We further took 5 random subsamples at every 5 participants, starting from 1, and built 3 random split-halves, again to keep computational cost manageable. A ceiling was estimated for each participant, and each participant's raw values were divided by this ceiling. (Note that this approach is even more conservative than the leave-one-out approach, because split-half correlations tend to be higher than one-vs.-rest, due to a reduction in noise when averaging (for each half).)

8. Language Modeling. To assess the models' performance on the normative next-word-prediction task, we used a dataset of 720 Wikipedia articles, WikiText-2 [32], with 2M training, 218k validation, and 246k test tokens (words and word-parts). These tokens were processed by model-specific tokenization with a maximum vocabulary size of 250k, selected based on the tokens' frequency in the model's original training dataset, and split up into blocks of 32 tokens each (both the vocabulary size and the length of blocks were constrained by computational cost limitations). We sequentially fed the tokens into models as explained in Methods_5 (Computational Models) and captured representations at each step from each model's final layer (penultimate layer before the classifier if the model has a readout). To predict the next word, we fit a linear decoder from those representations to the next token over words in the vocabulary (n=50k), on the training tokens. This decoder is trained with a cross-entropy-loss $L = -\sum_c^C t_c^i \log\left(\frac{e^{s_c^i}}{\sum_d^C e^{s_d^i}}\right)$ where t_c^i is the true label for class c and sample i, and s_c^i is the predicted probability of that class; the linear weights are updated with AdamW and a learning rate of 5e-5 in batches of 4 blocks until convergence as defined on the validation set. Importantly, note that we only trained weights of a readout decoder, *not* the weights of models themselves, in order to maintain the same model representations that we used in model-to-brain and model-to-behavior comparisons. The final language modeling score is reported for each model as the perplexity, i.e. the exponent of the cross-entropy loss, on the held-out test set. We ensured that our pipeline could reproduce the lower perplexity values in e.g. [31] by fine-tuning the entire model and increasing the batch size. To be able

to test all models under the same conditions and with fixed representations that were used for brain prediction, we however had to use a lower batch size and only train a linear readout without fine-tuning which leads to the lower perplexity scores reported in Fig. 3. T5-11b is not part of this analysis because of lack of computational resources to run the model.

9. Statistical tests. As a primary metric, model-to-brain predictivity scores are reported as the Pearson correlation coefficient (denoted by "r"). These correlation scores were obtained from aggregating over individual per-voxel/electrode/ROI scores. To avoid the assumption that the neural scores are Gaussian distributed, we aggregated these per-voxel/electrode/ROI scores by taking the *median* of scores for each participant's voxels/electrodes/ROIs and then computing the median and median absolute deviation (m.a.d.) across participants.

In addition to reporting an aggregated score across datasets, we show individual scores per dataset (visualized as bar plot insets). To obtain an error estimate for the correlation scores, we report the bootstrapped correlation coefficient, as computed by leaving out 10% of the scores and computing the r-value on the remaining 90% held-out scores (over 1,000 iterations).

All p-values less than 0.05 are summarized with one asterisk, p-values less than 0.005 with two asterisks, p-values less than 0.0005 with three asterisks, and p-values less than 0.00005 are denoted by four asterisks.

For interaction tests, we used two-sided t-tests with 1,000 bootstraps and 90% of samples per bootstrap.

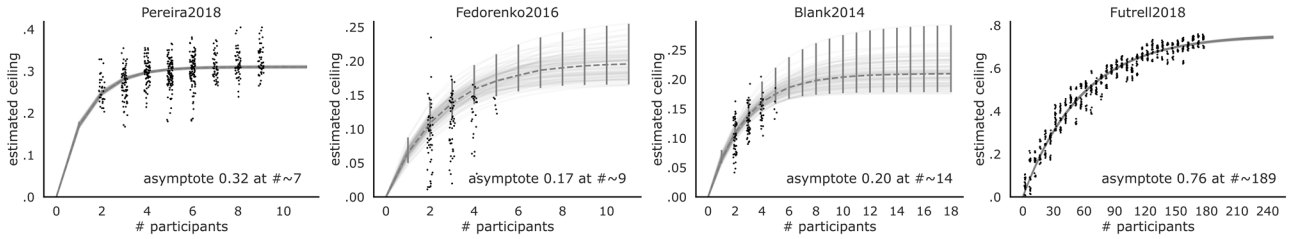

Figure S1: **Ceiling estimates for neural and behavioral datasets.** Due to intrinsic noise in biological measurements, we estimated a ceiling value to reflect how well the best possible model of an average human could perform, based on sub-samples of the total set of participants (see Methods-7). For each sub-sample, $s-1$ participants are used to predict a held-out participant (except in *Futrell2018*, where this is done on split-halves, as described in the text). Each dot represents a correlation between the average scores of the $s-1$ participants and the left-out participant for a random sub-sample of the number of participants s indicated on the x-axis. We then bootstrapped 100 random combinations of those dots to extrapolate (gray lines) the highest possible ceiling if we had an infinite number of participants at our disposal. The parameters of these bootstraps are then aggregated by taking the median to compute an overall estimated ceiling (dashed gray line with 95% CI in error-bars). We use this estimated ceiling to normalize model scores and here also report the number of participants at which the estimated ceiling would be met (which show that for *Pereira2018* and *Futrell2018,* the number of participants we have is at and close to the asymptote value, respectively). Ceiling levels are .32 (*Pereira2018*), .17 (*Fedorenko2016*), .20 (*Blank2014*), and .76 (*Futrell2018*).

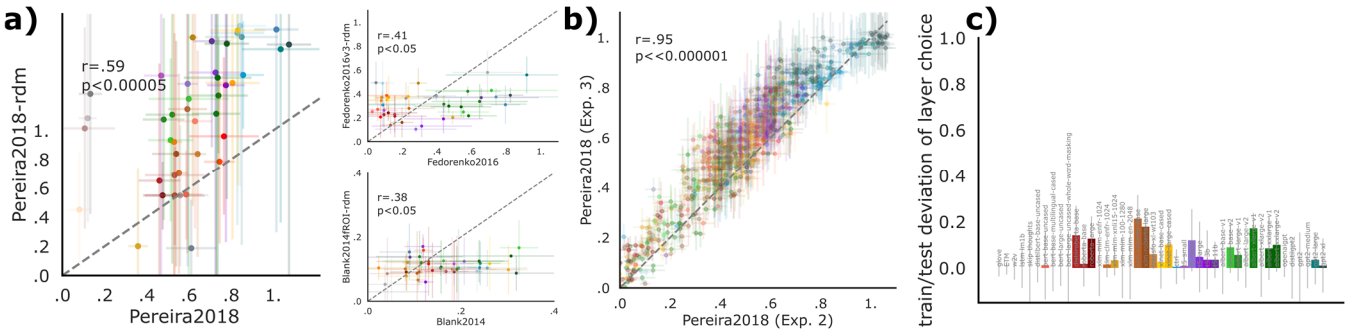

Figure S2: **Scores generalize across metrics and layers. a)** Model scores on each dataset generalize across different choices of a similarity metric; here we plot the predictivity metric used in the manuscript on the x-axis against a model-to-brain similarity metric based on representational dissimilarity matrices (RDMs) between models and neural representations on the y-axis. Like in the predictivity metric, stimuli along with corresponding model activations and brain recordings were split 5-fold but we then only compared the respective test splits given that the RDM metric does not employ fitting. Specifically, we followed [33] and computed the RDM for each model's activations, and a separate RDM for each brain recording dataset, based on 1 minus the Pearson correlation coefficient between pairs of stimuli; then, we measured model-brain similarity via Spearman correlation across the two RDMs' upper triangles. The RDM score for one model on one human dataset is then the mean over splits. We ran each model and compared resulting scores with the primarily used scores from the predictivity metric. Correlations for models' scores between the predictivity and the RDM metrics are: Pereira2018 r=.57, p<0.0001; Fedorenko2016 r=.40, p<.01; Blank2014 r=.38, p<.05. **b)** Model scores per layer generalize across dataset splits; for every layer in each model we plot its brain score (using the predictivity metric) on two experimental splits (experiment 2 and 3) of the *Pereira2018* dataset. Scores are very strongly correlated (r=.95, p<<0.000001), indicating that choosing a model's layer on a separate dataset split will generalize to a held-out test split. **c)** Choice of layer generalizes across dataset splits; for each model we plot the difference between its score on *Pereira2018* experiment 3 when choosing the layer on experiment 3 directly (i.e. the max due to layer choice on "test set") and its score on experiment 3 when choosing the layer on experiment 2 (choice on "train set"). The layer is chosen based on the model's maximum score across layers on the respective dataset split. Deviations between choosing the layer on a train or test set are minimal with error bars overlapping 0, indicating that there is no substantial difference between the two choices.

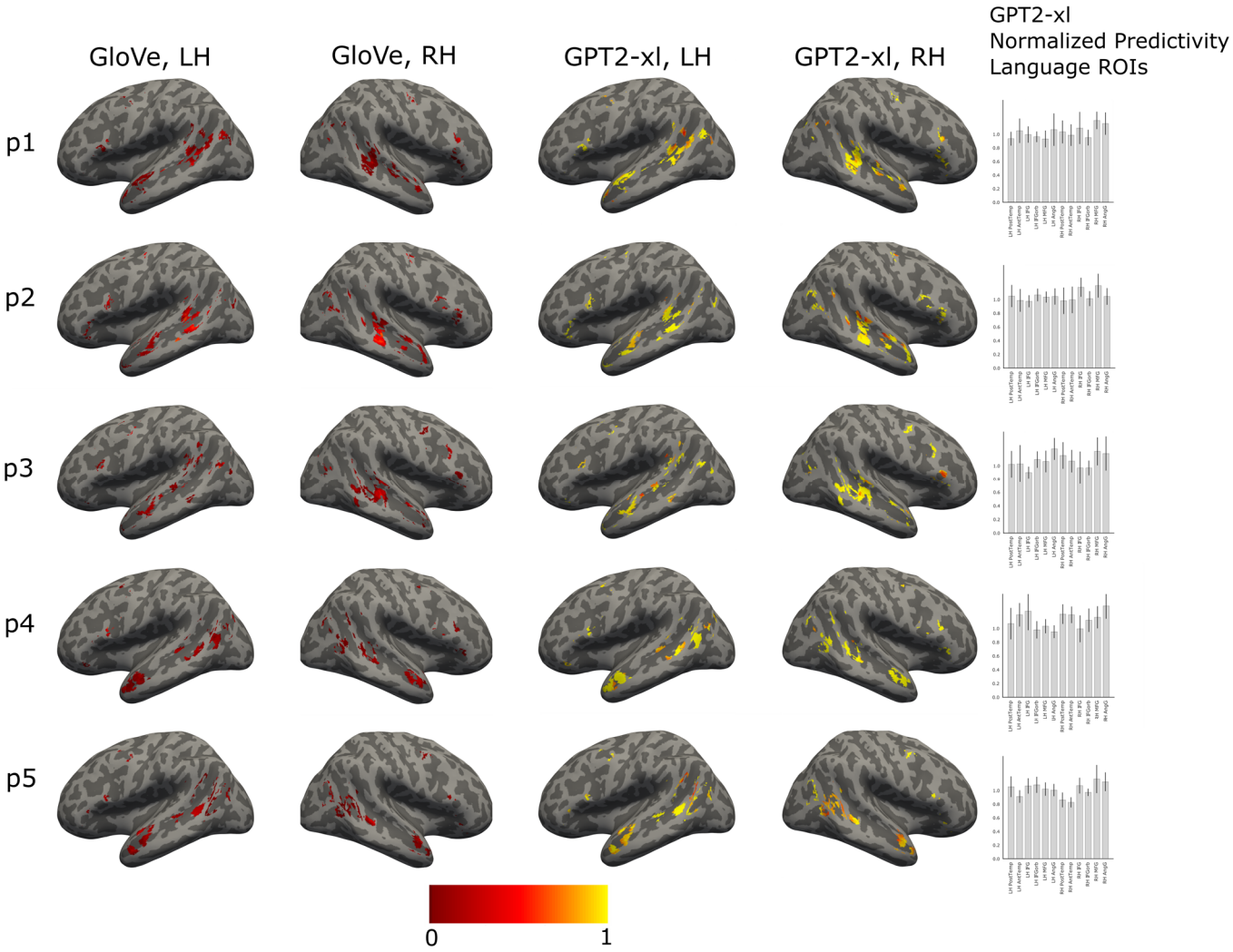

Figure S3: **Brain surface visualization of model predictivity scores.** Plots show surface projections of volumetric individual language-responsive functional ROIs in the left and right hemispheres (LH and RH) for five representative participants from *Pereira2018.* In each voxel of each fROI, we show a normalized predictivity value for two models that differ substantially in their ability to predict human data: GloVe (first two columns) and GPT2-xl (second two columns; for GPT2-xl, we show predictivity values from the overall best-performing layer, in line with how we report the results in the main text). (Note that the voxel locations are identical between GloVe and GPT2-xl, and are determined by an independent functional language localizer as described in the text; we here illustrate the differences in predictivity values, along with showing sample fROIs used in our analyses). Predictivity values were ceiling-normalized for each participant and each of 12 ROIs separately (a slight deviation from the approach in the main analysis, which was designed to control for between-region differences in reliability). The data were analyzed in the volume space and co-registered using SPM12 to Freesurfer's standard brain CVS35 (combined volumetric and surface-based (CVS)) in the MNI152 space using nearest neighbor interpolation and no smoothing. The ceiled predictivity maps for the language localizer contrast (10% of most language-responsive voxels in each 'mask'; Methods-1) were projected onto the cortical surface using mri_vol2surf in Freesurfer v6.0.0 with a projection fraction of 1. The surface projections were visualized on an inflated brain in the MNI152 space using the developer version of Freeview (assembly March 10th, 2020). The bar plots in the rightmost column show the normalized predictivity values per ROI (median across voxels) in the language network for GPT2-xl. Error bars denote m.a.d. across voxels. The distribution of predictivity values across the language-responsive voxels, and the similar predictivity magnitudes across the ROIs in the bar graphs, both suggest that the results (between-model differences in neural scores) are not driven by one particular region of the language network, but are similar across regions, and between the LH and RH components of the network (see also SI-4).

SI-1 – Language specificity

In the analyses reported in the manuscript, we focused on the language-responsive regions / electrodes. Here, for two datasets, we investigated the model-brain relationship outside the language network in order to assess the spatial specificity of our results, i.e., to test whether they obtain only, or more strongly, in the language network compared to other parts of the brain. For both datasets, we report analyses based on *raw predictivity values*, without normalizing by the estimated noise ceiling because the brain regions of the language network differ from other parts of the brain in how strongly their activity is tied to stimulus properties during comprehension (e.g., I. A. Blank & Fedorenko, 2017, 2020; Diachek et al., 2020; Shain et al., 2020; Wehbe et al., 2020). This variability is important to take into account when comparing between functionally different brain regions/electrodes because we are interested in how well the models explain linguistic-stimulus-related neural activity. When we normalize the neural responses of a non-language-responsive region/electrode using a language comprehension task, we're effectively isolating whatever little *stimulus-related activity* this region/electrode may exhibit, putting them on ~equal or similar footing with the language-responsive regions/electrodes. (For completeness and ease of comparison with the main analyses, we also report analyses based on normalized predictivity values.)

Fedorenko2016: The scores obtained from language-responsive electrodes were compared to those obtained from stimulus-responsive electrodes, excluding the language-responsive ones (see Methods-2), for all 43 models. The number of language-responsive electrodes across five participants was 97, and the number of stimulus-, but not language-, responsive electrodes across the participants was comparable (n=105). The analysis was identical to the main analysis (see Methods), besides omitting the ceiling normalization for the raw predictivity analyses. As described in Methods, normalization was performed for each electrode in each participant separately.

For raw predictivity, neural responses in the language-responsive electrodes were predicted 49.21% better on average across models than the non-language-responsive electrodes (independent-samples two-tailed t-test: t=3.4, p=0.001). (For normalized predictivity, neural responses in the language-responsive electrodes were predicted 59.26% better on average across models than the non-language-responsive electrodes (t=2.24, p=0.03).)

Pereira2018: The scores obtained from the language network were compared to those obtained from two control networks: the multiple demand (MD) network and the default mode network (DMN) (see Methods), for all 43 models. The number of voxels in the language network across participants was, on average, 1,355 (± 7 SD across participants), and the average number of voxels in the MD network and the DMN was comparable (MD: 2,994±230); DMN: 1,098±7). The analysis was identical to the main analysis (see Methods), besides omitting the ceiling normalization for the raw predictivity analyses. For the normalized predictivity analyses, the network predictivity values were normalized by their respective network ceiling values.

For raw predictivity, neural responses in the language network ROIs were predicted 16.96% better on average across models than the MD network ROIs (independent-samples two-tailed t-test: t=2.26, p=0.03) and numerically (14.33%) better than the DMN ROIs (t=1.78, p=0.08). (For normalized predictivity, neural responses in the language network ROIs were predicted numerically (6.47%) worse on average than the MD network ROIs (t=-0.92, p=0.36) and also numerically (1.05%) worse than the DMN ROIs (t=-0.31, p=0.76).)

These results suggest that—when allowing for inter-regional differences in the reliability of language-related responses—the model-to-brain relationship is stronger in the language-responsive regions/electrodes. However, we leave open the possibility that language models also explain neural responses outside the boundaries of the language network, perhaps because these models capture some parts of our general semantic knowledge, which is plausibly stored in a distributed fashion across the brain. For example, several earlier studies used simple embedding models to decode linguistic meaning from fMRI data (e.g., Wehbe et al., 2014; Huth et al., 2016; Anderson et al., 2017; Pereira et al., 2018) and reported reliable decoding not only within the language network, but also across other parts of association cortex. Given that we know that different large-scale cortical networks differ functionally in important ways (e.g., see Fedorenko & Blank, 2020, for a recent discussion of the language vs. MD networks), it will be important to investigate in future work the precise mapping between the language models' representations and neural responses in these different functional networks.

SI-2 – Model performance on diverse language tasks vs. model-to-brain fit

To test whether the next-word prediction task is special in predicting model-to-brain fit, we used the *Pereira2018* dataset to examine the relationship between the models' performance on diverse language processing tasks from the General Language Understanding Evaluation (GLUE) benchmarks (Wang et al., 2018) and neural predictivity. We used a subset of the high-performing, transformer models (n=30 of the 38 where we could find published commitments of which features to use for GLUE). The GLUE benchmark encompasses nine tasks that can be classified into three categories: single-sentence judgment tasks (n=2), sentence-pair semantic similarity judgment tasks (n=3), and sentence-pair inference tasks (n=4). The two single-sentence tasks are both binary classification tasks: models are asked to determine whether a given sentence is grammatical or ungrammatical (Corpus of Linguistic Acceptability, *CoLA* (Warstadt et al., 2018)), or whether the sentiment of a sentence is positive or negative (Stanford Sentiment Treebank, *SST-2* (Socher et al., 2013)). In the semantic similarity tasks, models are asked to assert or deny the semantic equivalence of question pairs (Quora Question Pairs, *QQP* (Chen et al., 2018)) or sentence pairs (Microsoft Research Paraphrase Corpus, *MRPC* (Dolan & Brockett, 2005)), or to judge the degree of semantic similarity between two sentences on a scale of 1-5 (Semantic Textual Similarity Benchmark, *STS-B* (Cer et al., 2017)). Lastly, the benchmark contains four inference tasks, of which we include three (following Devlin et al., 2018), we exclude the Winograd Natual Language Inference, *WNLI*, task; see (12) in https://gluebenchmark.com/faq). In two of these tasks, models are asked to determine the entailment relationship between sentences in a pair using either tertiary classification: entailment, contradiction, neutral (Multi-Genre Natural Language Inference corpus, *MNLI* (Williams et al., 2018)), or binary classification: entailment or no entailment (Recognizing Textual Entailment, *RTE* (Dagan et al., 2006, Bar Haim et al., 2006, Giampiccolo et al., 2007, Bentivogli et al., 2009)). And in the third inference task, the Question Natural Language Inference, *QNLI,* task (Rajpurkar et al., 2016, White et al., 2017, Demszky et al., 2018), models are presented with question-answer pairs and asked to decide whether or not the answer-sentence contains the answer to the question.

In order to evaluate model performance on GLUE benchmark tasks, each GLUE dataset was first converted into a format that is compatible with transformer model input using functionality from the GLUE data processor provided by Huggingface transformers (https://huggingface.co/transformers/). In particular, each set of materials is represented as a matrix that includes the following dimensions: item (and sentence for multi-sentence materials) ID, ID for each individual word (with reference to the vocabulary used by the transformer models), the label (e.g., grammatical vs. ungrammatical), and the 'attention mask' which specifies which part(s) of the sentences the model should pay attention to (e.g., some 'padding' is commonly used to equalize the lengths of sentences/items to the target length of 128 tokens (again constrained by computational cost), and the attention mask is set to include only the actual words in the materials, and not the padding, and in some models to further constain which parts of the input to attend to—e.g., in GPT2 models, the rightward context is ignored). Next, each GLUE dataset was then fed into each model to obtain a sequence of hidden states at the output of the last layer of the model. Following default settings from Huggingface transformers, from these hidden states, we then extracted the token of interest: for bidirectional models such as BERT, this was the first input token—a special token ([cls]) that is appended to each item and designed for sequence classification tasks, and for unidirectional models such as GPT-2, XLNet or CTRL, this token corresponded to the last attended token (e.g., the last word/word-part in the sentence). In order to ensure a fair comparison between the models and to avoid the skewing of representations by individual task pre-training, dense linear pooling projection layers (specific to some transformer) are disregarded. Finally, we fit a linear decoder from the features of the extracted tokens of interest to the task label(s). For tasks with two or more labels, a cross-entropy loss function is used; for the task that uses a rating scale, the decoder is trained with a mean-square error (MSE) loss function. Similar to the next-word prediction task, the linear weights are updated with the AdamW optimizer and a learning rate of 5e-5 in batches of 8 blocks until convergence as defined on the validation set. Importantly, and also similar to the next-word-prediction task, we only trained weights of a readout decoder, *not* the weights of models themselves, in order to maintain the same model representations that we used in model-to-brain and model-to-behavior comparisons. To account for potential bias in the GLUE datasets, multiple metrics within tasks, as well as different metrics across tasks are reported in the GLUE benchmark. Following standards in the field, we follow GLUE evaluation metrics [27] and report the final task score as accuracy for *SST-2, MNLI*, *RTE,* and *QNLI*, Matthew's Correlation for *CoLA*, the average of accuracy and F1 score for MRPC, and QQP, and the

average of Pearson and Spearman correlation for *STS-B*. The results are shown in Fig. S5. None of the tasks significantly predicted neural scores, suggesting that next-word prediction may be special in its ability to predict brain-like processing. As with language modeling, we were unable to evaluate T5-11b on these benchmarks due to lack of computational resources.

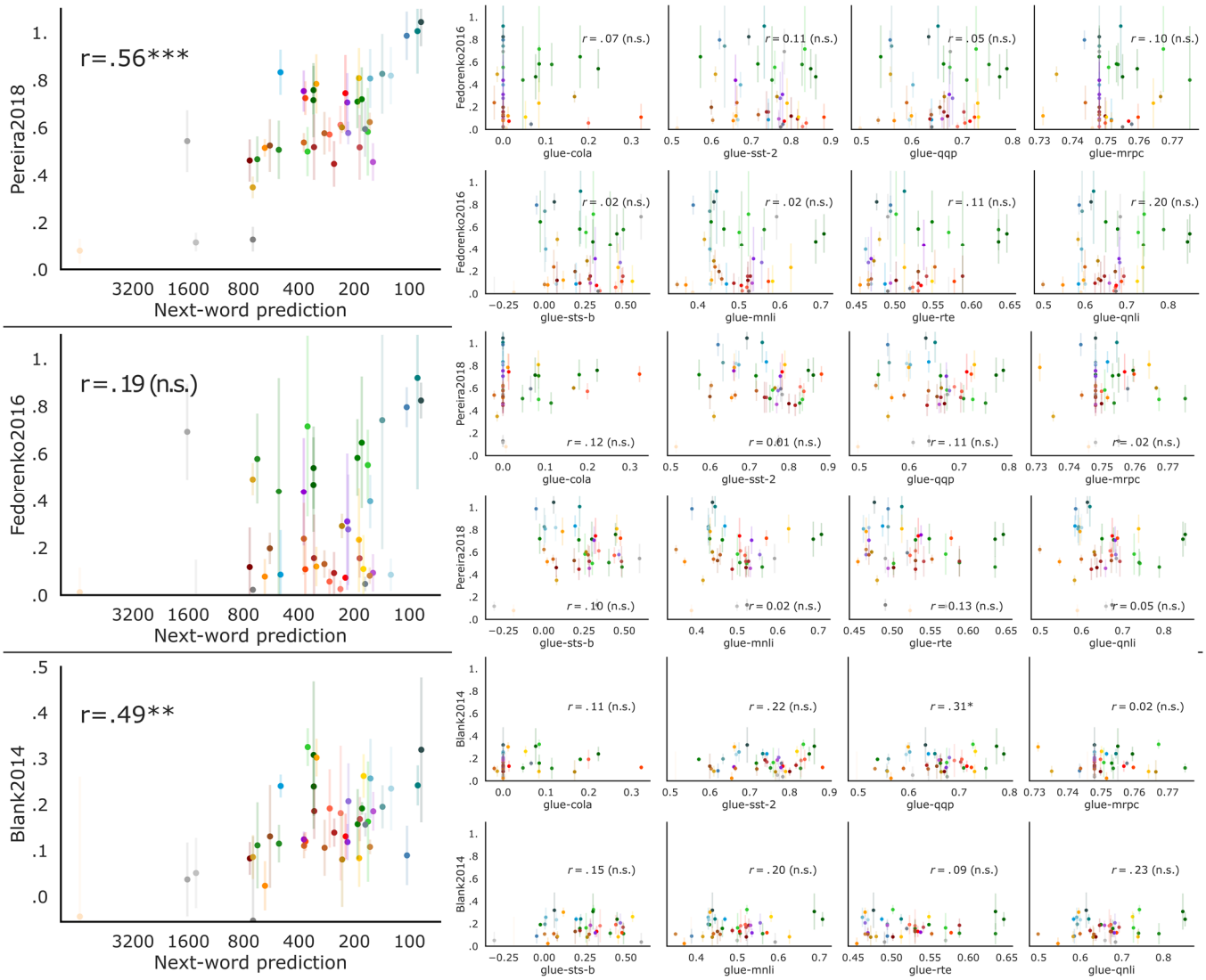

Figure S4: **Performance on next-word prediction selectively predicts model-to-brain fit**. Performance on GLUE tasks was evaluated as described in SI-5. Only the next-word prediction correlations but none of the GLUE correlations were significant.

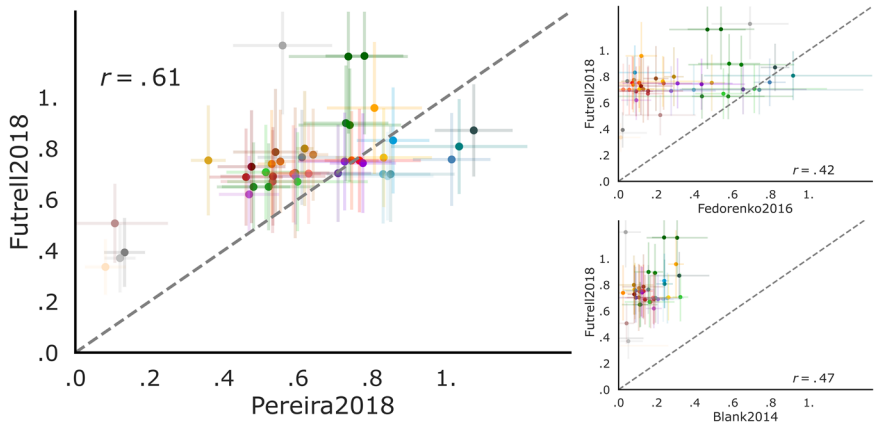

Figure S5: **Models' neural predictivity for each dataset is correlated with behavioral predictivity.** In Fig. 4b, we showed that the models' neural predictivity (averaged across the three neural datasets: Pereira2018, Fedorenko2016, Blank2014) correlates with behavioral predictivity. Here, we show that this relationship also holds for each neural dataset individually: Pereira2018: p<0.0001, Fedorenko2016: p<0.01, Blank2014: p<0.01.

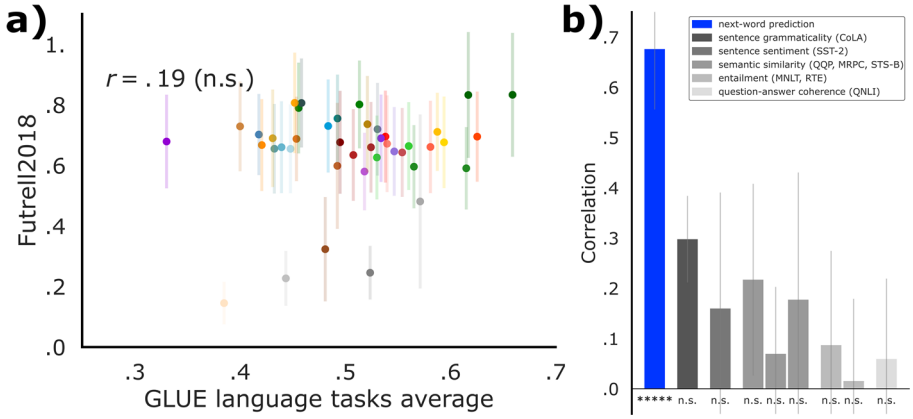

Figure S6: **Performance on GLUE tasks does *not* predict model-to-behavior fit**. In Fig. 4c, we showed a significant positive correlation of next-word prediction performance with predictivity on behavioral reading times. Here we test whether performance on GLUE tasks predicts behavioral scores (performance on GLUE tasks was evaluated as described in SI-5). Only the next-word prediction correlations but none of the GLUE correlations were significant. Notations as in Figure 3 for the GLUE average (a) and individual tasks (b).

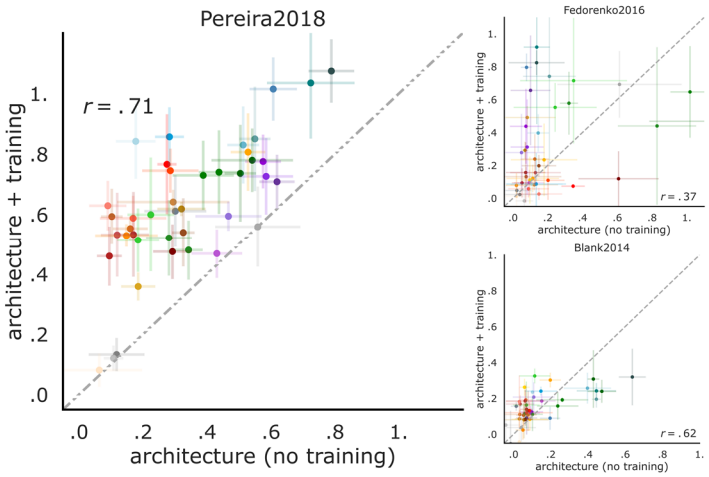

Figure S7: **Model architecture contributes to brain predictivity and untrained performance predicts trained performance.** In Fig. 5, we showed that untrained models already achieve robust brain predictivity (averaged across the three neural and one behavioral datasets). Here, we show that this relationship also holds for each dataset individually: Pereira2018: $p \ll 0.00001$, Fedorenko2016: $p < 0.05$, Blank2014: $p < 0.00001$.

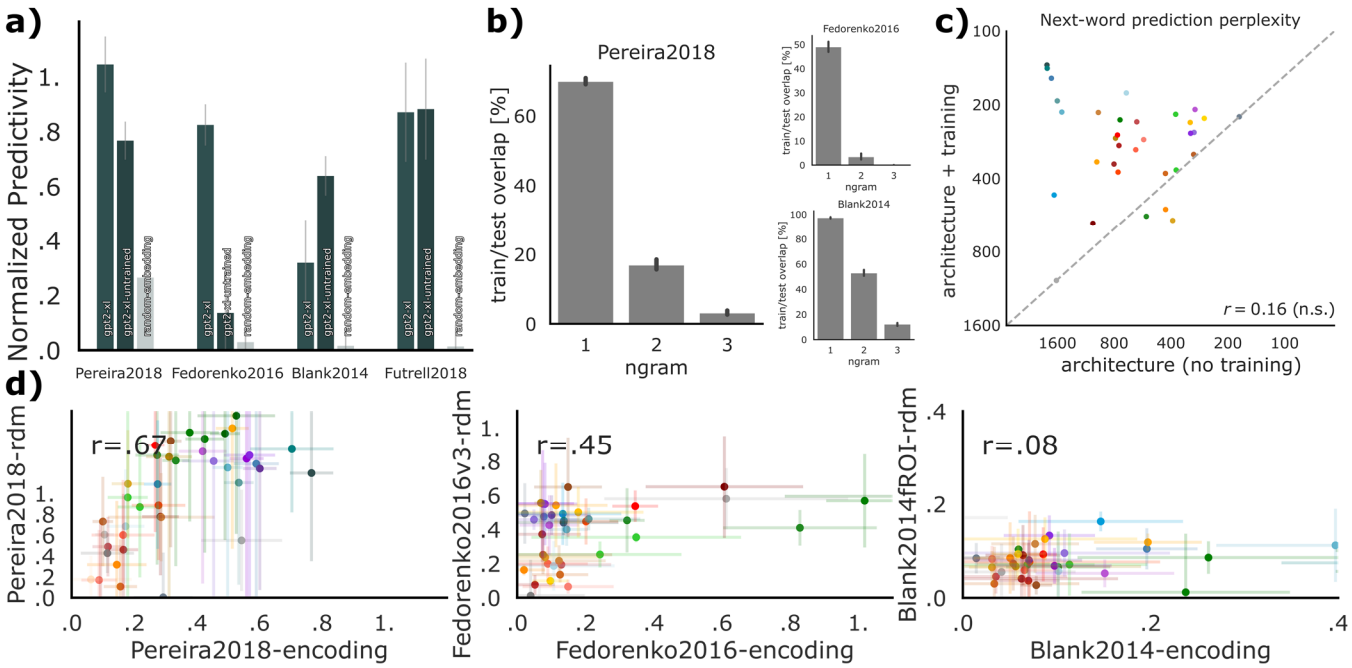

Figure S8: **Controls for untrained models. a)** Neural and behavioral scores of GPT2-xl, the best-performing model, with vs. without training, and of a random embedding of the same size. A large feature size alone is not sufficient: a random embedding matched in size to GPT2-xl scores worse than untrained GPT2-xl in all four datasets (3 neural, and 1 behavioral). These results suggest that model architecture critically contributes to model-to-brain and model-to-behavior fits. **b)** Overlap of bi- and tri-grams in train/test stimuli splits of benchmarks is minimal, and despite single-word overlap memorization of per-word responses is insufficient (a). **c)** The relationship between model performance with vs. without training on the wikitext-2 next-word-prediction task. Consistent with model performance with vs. without training on neural and behavioral datasets (Fig. 5), untrained models perform reasonably well. Training improves scores by 80% on average, and most prominently for GPT models, in teal (where the quality of the training data is optimized; see Computational models in Methods). GPT's poor performance on next-word prediction might be explained by very high representational similarities across words pre-training in its last layer [38]. **d)** Scores for untrained models obtained via linear predictivity generalize to scores obtained via RDM correlations. The RDM metric does not use any fitting. Correlations for untrained models' scores between the predictivity and the RDM metric are: Pereira2018 r=.67, p<0.000005; Fedorenko2016 r=.45, p<.005; Blank2014 r=.08, n.s. See Fig. S2 for details on the RDM metric.

SI-3 – Effects of model architecture and training on neural and behavioral scores

The 43 language models included in the current study span three major types of architecture: embedding models, recurrent models, and attention-based transformer architectures. However, in addition to this coarse distinction, the individual models vary widely in diverse architectural and training features. A rigorous examination of the effects of different model features on model-to-brain/behavior fit would require careful pairwise comparisons of minimally different models, which is not possible for 'off-the-shelf' models without extremely expensive re-training from scratch under many/all possible combinations of architecture, training diet, optimization objective, and other hyper-parameters. However, we here undertook a preliminary exploratory investigation. In particular, for a subset of model features (Table SI-9), we computed a Pearson correlation between the feature values and the averaged model score across all four datasets (3 neural, and 1 behavioral). We included five architectural features. Three features were continuous: i) number of hidden layers, which varied between 1 and 48 (mean 16.02, std. dev. 11.02); ii) number of features (units across considered layers), which varied between 300 and 78,400 (mean 20,971.26, std. dev. 18,362.91); and iii) the size of the embedding layer, which varied between 128 and 48,000 (mean 872.28, std. dev. 744.33). And the remaining two features were binary: iv) uni- vs. bi-directionality (32/43 models were bi-directional), and v) the presence of recurrence (5/43 models had recurrence). And we included two training-related features: i) training data size (in GB), which varied between 0.2 and 336 (mean 351.06 std. dev. 726.81); and ii) vocabulary size, which varied between 30,000 and 3,000,000 (mean 223,096.95 std. dev. 561,737.36). All training data numbers were taken from the original model papers, and if training data was specified in tokens, a conversion rate of 4 bytes per token was used. We further excluded the multilingual XLM and BERT models when examining the effect of training data size, because those numbers could not be confidently verified. For comparison, we also included performance on the next-word-prediction task that we examined in the main text.

The results are shown in Fig. S10. As expected—given the results reported in the main text for the individual datasets (Fig. 3, 4c)—next-word prediction performance robustly predicts model-to-brain/behavior fit ($r = 0.49$, $p < 0.01$). These results suggest that optimizing for predictive representations may be a critical shared feature of biological and artificial neural networks for language. How do architectural and training-related features compare to next-word-prediction task performance in their effect on neural/behavioral predictivity? Two architectural size features are most correlated with model performance: number of hidden layers ($r = 0.56$, $p < 0.001$), and number of features ($r = 0.68$, $p << 0.0001$). This is expected given that the most recent models with the highest performance on linguistic tasks are also the largest ones that researchers are able to run on modern hardware. The two training-related features—training data size and vocabulary size—are significantly *negatively* correlated with model performance. To rule out the possibility that the negative effect of training-related features is driven by models with relatively small training datasets and vocabulary size (e.g., ETM; Table S11) that have low brain/behavior predictivity, we ran an additional analysis considering only transformer models (n=38): even in these generally highly predictive models, more training data ($r = -0.29$, $p = 0.11$ [not plotted]) or larger vocabulary size ($r = -0.21$, $p = 0.25$ [not plotted]) do not appear to be beneficial, although the negative correlations are non-significant.

Does the collection of model designs investigated in this paper inform the hyperparameters that should be optimized for in any new model to achieve high predictivity? To provide a preliminary answer to this question, we performed an exploratory analysis in the form of stepwise forward model selection and examined (a) the most parsimonious model that explains the data, and (b) how much variance the selected features explain cumulatively (Fig. S10b). High overall explained variance indicates that the combination of features selected by the model is predictive of model performance, whereas low overall explained variance indicates that crucial predictive hyperparameters are still being neglected. In the forward regression analysis, we add predictors based on the highest R^2-adjusted value of the new model, as long as variance increases by adding a new factor. This analysis revealed that adding training dataset size and recurrence does not lead to variance increase. Significance markers indicate the p-value for significance of adding each term, and for each regression step we plot the added explained variance (in R^2-adjusted) of the variable chosen by the model. The overall cumulative R^{2-}adjusted value of the selected model is 0.822.

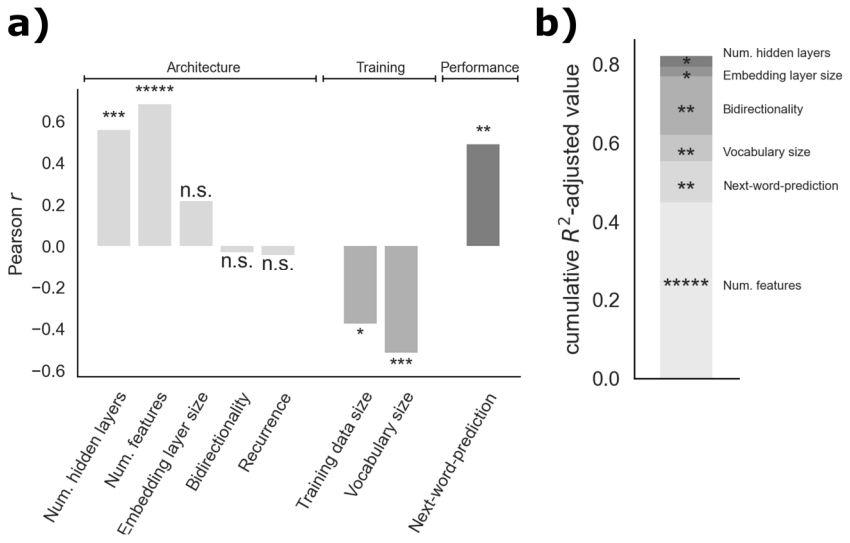

Figure S9: **Effects of model architecture vs. training on neural and behavioral scores. a)** We compared the effects on neural and behavioral scores (the averaged model score across all four datasets) of three kinds of features: (i) architectural properties, (ii) training-dependent variables, and, for comparison, iii) performance on the next-word-prediction task examined in the main text (Fig. 3, 4c). **b**) Alternative combination of predictors with stepwise forward regression model. New predictors are added based on the highest R^2-adjusted value of the new model, as long as variance increases by adding a new factor (thus excluding training dataset size and recurrence). Significance markers indicate the p-value for significance of adding model terms. For each regression step, we plot the added explained variance (in R^2-adjusted) of the variable chosen by the model. The overall cumulative R^{2}-adjusted value of the selected model is 0.822. As in a), the preferred explanatory variable is the number of features. Stepwise forward regression based on significance leads to the same model-choice. Note that, as above, t5-11b is excluded for regression based on next-word-prediction, and multilingual models are excluded for regression on training size.

	Model identifier	Architecture class	Num. layers	Num. features	Embedding layer size	Bidirectional	Recurrent	Training data size	Vocabulary size	Tokenization	Training tasks
1	glove	Embedding	1	300	300	0	0	3360	2200000	Stanford tokenizer	Learning word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence
2	ETM	Embedding	1	300	300	0	0	0.2	52258	RegEx word-level tokenizer	Variational inference topic modeling using embedding representations of both words and topics
3	word2vec	Embedding	1	300	300	0	0	400	3000000	Word-level tokenizer	Predicting a center word from the surrounding context
4	lstm lm_1b	Recurrent	2	2048	1024	0	1	4	793471	bBPE	Causal Language Modeling
5	skip-thoughts	Recurrent	1	4800	4800	0	1	3	930911	NLTK tokenizer	Predicting words in neighboring sentences
6	distilbert-base-uncased	Bidir. transf.	6	5376	768	1	0	13	30522	WordPiece	Masked Language Modeling Next-Sentence Prediction
7	bert-base-uncased	Bidir. transf.	12	9984	768	1	0	13	30522	WordPiece	
8	bert-base-multilingual-cased	Bidir. transf.	12	9984	768	1	0	n.a.	119547	WordPiece	
9	bert-large-uncased	Bidir. transf.	24	25600	1024	1	0	13	30522	WordPiece	
10	bert-large-uncased-whole-word-masking	Bidir. transf.	24	25600	1024	1	0	13	30522	WordPiece	
11	distilroberta-base	Bidir. transf.	6	5376	768	1	0	161	50265	bBPE	dynamic Masked Language Modeling
12	roberta-base	Bidir. transf.	12	9984	768	1	0	161	50265	bBPE	
13	roberta-large	Bidir. transf.	24	25600	1024	1	0	161	50265	bBPE	
14	xlm-mlm-enfr-1024	Bidir. transf.	6	7168	1024	1	0	n.a.	64139	BPE	multilingual Masked Language Modeling
15	xlm-clm-enfr-1024	Bidir. transf.	6	7168	1024	1	0	n.a.	64139	BPE	multilingual Causal Language Modeling
16	xlm-mlm-xnli15-1024	Bidir. transf.	12	13312	1024	1	0	n.a.	95000	BPE	multilingual Masked Language Modeling
17	xlm-mlm-100-1280	Bidir. transf.	16	21760	1280	1	0	n.a.	200000	BPE	
18	xlm-mlm-en-2048	Bidir. transf.	12	26624	2048	1	0	16	30145	BPE	Masked Language Modeling
19	xlm-roberta-base	Bidir. transf.	12	9984	768	1	0	2500	250002	SentencePiece	multilingual Masked Language Modeling
20	xlm-roberta-large	Bidir. transf.	25	25600	1024	1	0	2500	250002	SentencePiece	
21	transfo-xl-wt103	Bidir. transf.	18	19456	1024	1	1	0.4	267735	Word-level tokenizer	Causal Language Modeling
22	xlnet-base-cased	Bidir. transf.	12	9984	768	1	1	126	32000	SentencePiece	Permutation Language Modeling
23	xlnet-large-cased	Bidir. transf.	24	25600	1024	1	1	126	32000	SentencePiece	
24	ctrl	Bidir. transf.	48	62720	1280	1	0	140	246534	BPE	Causal Language Modeling
25	t5-small	Bidir. transf.	6	3584	512	1	0	862	32128	SentencePiece	Text-to-text training on a variety of tasks (i.a., prediction of multiple corrupted tokens, and tasks from the GLUE and SuperGLUE benchmarks)
26	t5-base	Bidir. transf.	12	9984	768	1	0	862	32128	SentencePiece	
27	t5-large	Bidir. transf.	24	25600	1024	1	0	862	32128	SentencePiece	
28	t5-3b	Bidir. transf.	24	25600	1024	1	0	862	32128	SentencePiece	
29	t5-11b	Bidir. transf.	24	25600	1024	1	0	862	32128	SentencePiece	
30	albert-base-v1	Bidir. transf.	12	9984	128	1	0	16	30000	SentencePiece	Masked Language Modeling Sentence-Order Prediction
31	albert-base-v2	Bidir. transf.	12	9984	128	1	0	16	30000	SentencePiece	
32	albert-large-v1	Bidir. transf.	24	25600	128	1	0	16	30000	SentencePiece	
33	albert-large-v2	Bidir. transf.	24	25600	128	1	0	16	30000	SentencePiece	
34	albert-xlarge-v1	Bidir. transf.	24	51200	128	1	0	16	30000	SentencePiece	
35	albert-xlarge-v2	Bidir. transf.	24	51200	128	1	0	16	30000	SentencePiece	
36	albert-xxlarge-v1	Bidir. transf.	12	53248	128	1	0	16	30000	SentencePiece	
37	albert-xxlarge-v2	Bidir. transf.	12	53248	128	1	0	16	30000	SentencePiece	
38	openaigpt	Unidir. transf.	12	9984	768	0	0	3	40478	BPE	Causal Language Modeling
39	distilgpt2	Unidir. transf.	6	5376	768	0	0	40	50257	bBPE	Causal Language Modeling
40	gpt2	Unidir. transf.	12	9984	768	0	0	40	50257	bBPE	
41	gpt2-medium	Unidir. transf.	24	25600	1024	0	0	40	50257	bBPE	
42	gpt2-large	Unidir. transf.	36	47360	1280	0	0	40	50257	bBPE	
43	gpt2-xl	Unidir. transf.	48	78400	1600	0	0	40	50257	bBPE	

Table S1: **Overview of model designs**.

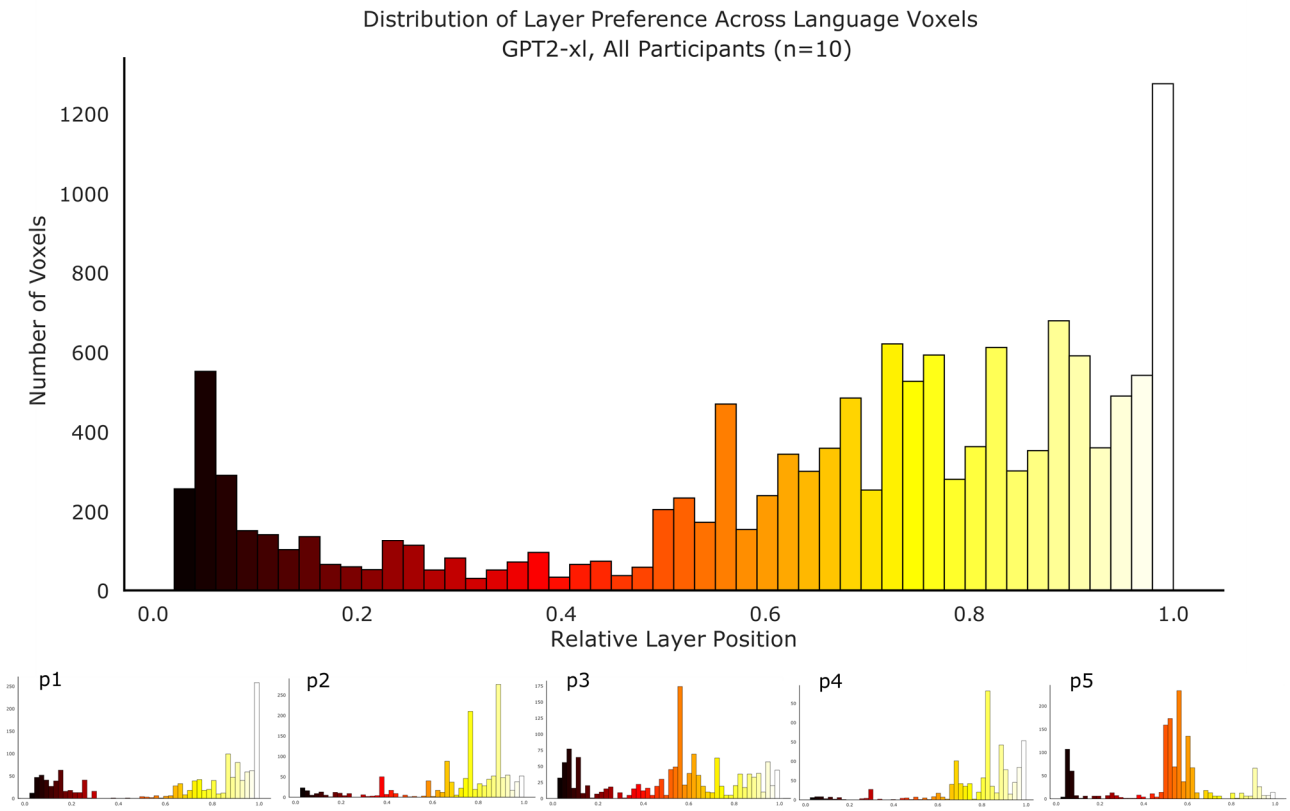

Figure S10: **Distribution of layer preference (best performing layer) per voxel for GPT2-xl for *Pereira2018***. A per-voxel per-participant raw predictivity value (as opposed to *overall* ceiled predictivity scores in Fig. 2c) was obtained in the language network by computing the mean over cross-validation splits and experiments. For each voxel, the layer with the highest predictivity value was estimated as the "preferred" layer (argmax over layer scores). As in the main analyses, the voxels in the language network were included. Zero on the x-axis corresponds to the embedding layer of the model. The upper plot is averaged across all participants in *Pereira2018* (n=10). The lower panel shows the participant-wise layer preference for five representative participants. Across participants, most voxels show the highest predictivity value for later layers of GPT2-xl. Within participants, the layer preference across voxels varies but is often clustered around particular layers. Investigations of how predictivity fluctuates across model layers, and/or between the language network and other parts of the brain, is left for future work.

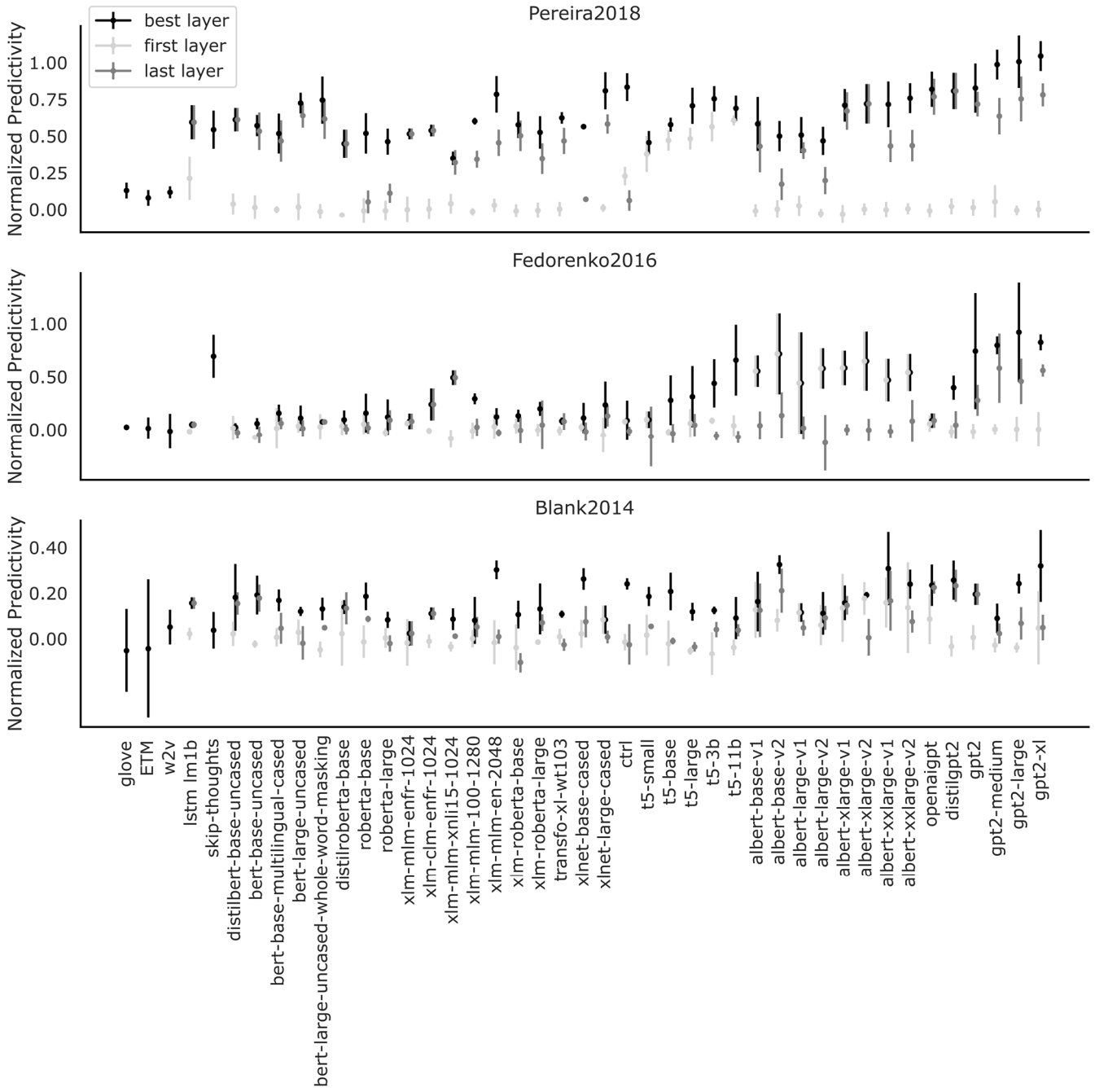

Figure S11: **Brain scores of each model's best, first, and last layer**. To test the importance of intermediate representations, we directly compared layer performances at the beginning and end of each model with the model's best-performing layer. In nearly all networks with multiple layers, both the token embedding (first layer) as well as the task-specific output (last layer) underperform significantly compared to the respective best layer. This suggests that the combination of architecture and weights in the networks is a major driver for brain-like representations, beyond potential semantic information that is already present in the model input codes. Lexical similarity determined by optimizing for next-word prediction present in the output layer is also not sufficient, instead pointing to intermediate representations as the most predictive (see also Fig. 2c).

SI References

[1] A. Bautista and S. M. Wilson, "Neural responses to grammatically and lexically degraded speech," *Lang. Cogn. Neurosci.*, vol. 31, no. 4, pp. 567–574, Apr. 2016, doi: 10.1080/23273798.2015.1123281.

[2] I. Blank, Z. Balewski, K. Mahowald, and E. Fedorenko, "Syntactic processing is distributed across the language system," *Neuroimage*, vol. 127, pp. 307–323, Feb. 2016, doi: 10.1016/j.neuroimage.2015.11.069.

[3] E. Fedorenko, P. J. Hsieh, A. Nieto-Castañón, S. Whitfield-Gabrieli, and N. Kanwisher, "New method for fMRI investigations of language: Defining ROIs functionally in individual subjects," *J. Neurophysiol.*, vol. 104, no. 2, pp. 1177–1194, Aug. 2010, doi: 10.1152/jn.00032.2010.

[4] E. Fedorenko, A. Nieto-Castañon, and N. Kanwisher, "Lexical and syntactic representations in the brain: An fMRI investigation with multi-voxel pattern analyses," *Neuropsychologia*, vol. 50, no. 4, pp. 499–513, Mar. 2012, doi: 10.1016/j.neuropsychologia.2011.09.014.

[5] E. Fedorenko *et al.*, "Neural correlate of the construction of sentence meaning," *Proc. Natl. Acad. Sci. United States Am.*, vol. 113, no. 41, pp. E6256–E6262, Oct. 2016, doi: 10.1073/pnas.1612132113.

[6] E. Fedorenko, I. Blank, M. Siegelman, and Z. Mineroff, "Lack of selectivity for syntax relative to word meanings throughout the language network," *bioRxiv Prepr.*, 2020, doi: 10.1101/477851.

[7] I. A. Blank and E. Fedorenko, "No evidence for differences among language regions in their temporal receptive windows," *Neuroimage*, vol. 219, p. 116925, Oct. 2020, doi: 10.1016/j.neuroimage.2020.116925.

[8] G. Buzsáki, C. A. Anastassiou, and C. Koch, "The origin of extracellular fields and currents-EEG, ECoG, LFP and spikes," *Nat. Rev. Neurosci.*, vol. 13, no. 6, pp. 407–420, Jun. 2012, doi: 10.1038/nrn3241.

[9] S. Lawrence Marple, "Computing the discrete-time analytic signal via fft," *IEEE Trans. Signal Process.*, vol. 47, no. 9, pp. 2600–2603, Sep. 1999, doi: 10.1109/78.782222.

[10] M. A. Just, P. A. Carpenter, and J. D. Woolley, "Paradigms and processes in reading comprehension," *J. Exp. Psychol. Gen.*, vol. 111, no. 2, pp. 228–238, Jun. 1982, doi: 10.1037/0096-3445.111.2.228.

[11] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162.

[12] A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, "Topic Modeling in Embedding Spaces," *arXiv Prepr.*, Jul. 2019, Accessed: Jun. 15, 2020. [Online]. Available: http://arxiv.org/abs/1907.04907.

[13] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations ofwords and phrases and their compositionality," Oct. 2013, Accessed: Jun. 15, 2020. [Online]. Available: http://arxiv.org/abs/1310.4546.

[14] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the Limits of Language Modeling," Feb. 2016, Accessed: Nov. 15, 2018. [Online]. Available: http://arxiv.org/abs/1602.02410.

[15] C. Chelba *et al.*, "One billion word benchmark for measuring progress in statistical language modeling," in *Annual Conference of the International Speech Communication Association*, Dec. 2014, pp. 2635–2639, Accessed: Jun. 15, 2020. [Online]. Available: http://arxiv.org/abs/1312.3005.

[16] R. Kiros *et al.*, "Skip-Thought Vectors," in *Neural Information Processing Systems (NIPS)*, 2015, pp. 3294–3302, Accessed: Jul. 09, 2019. [Online]. Available: http://papers.nips.cc/paper/5950-skip-thought-vectors.

[17] T. Wolf *et al.*, "HuggingFace's Transformers: State-of-the-art Natural Language Processing," *arXiv Prepr.*, Oct. 2019, Accessed: Jun. 15, 2020. [Online]. Available: http://arxiv.org/abs/1910.03771.

[18] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv Prepr.*, Oct. 2019, Accessed: Jun. 15, 2020. [Online]. Available: http://arxiv.org/abs/1910.01108.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv Prepr.*, Oct. 2018, Accessed: Oct. 11, 2018. [Online]. Available: https://arxiv.org/abs/1810.04805.

[20] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv Prepr.*, Jul. 2019, Accessed: Jun. 15, 2020. [Online]. Available: http://arxiv.org/abs/1907.11692.

[21] G. Lample and A. Conneau, "Cross-lingual Language Model Pretraining," in *Neural Information Processing Systems (NeurIPS)*, Jan. 2019, pp. 7059–7069, Accessed: May 07, 2020. [Online]. Available: http://arxiv.org/abs/1901.07291.

[22] A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," *arXiv Prepr.*, Nov. 2019, Accessed: Jun. 15, 2020. [Online]. Available: http://arxiv.org/abs/1911.02116.

[23] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Association for Computational Linguistics (ACL)*, Jan. 2020, pp. 2978–2988, doi:

10.18653/v1/p19-1285.

[24] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," *arXiv Prepr.*, Jun. 2019, Accessed: Jul. 09, 2019. [Online]. Available: http://arxiv.org/abs/1906.08237.

[25] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "CTRL: A Conditional Transformer Language Model for Controllable Generation," *arXiv Prepr.*, Sep. 2019, Accessed: Jun. 15, 2020. [Online]. Available: http://arxiv.org/abs/1909.05858.

[26] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *arXiv Prepr.*, Oct. 2019, Accessed: May 07, 2020. [Online]. Available: http://arxiv.org/abs/1910.10683.

[27] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," Sep. 2019, Accessed: May 21, 2020. [Online]. Available: http://arxiv.org/abs/1804.07461.

[28] A. Wang *et al.*, "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems," in *Neural Information Processing Systems (NeurIPS)*, 2019, pp. 3266–3280, Accessed: Jun. 15, 2020. [Online]. Available: http://arxiv.org/abs/1905.00537.

[29] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," *arXiv Prepr.*, Sep. 2019, Accessed: Jun. 15, 2020. [Online]. Available: http://arxiv.org/abs/1909.11942.

[30] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," 2018. [Online]. Available: https://gluebenchmark.com/leaderboard.

[31] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," *arXiv Prepr.*, 2019, [Online]. Available: https://github.com/codelucas/newspaper.

[32] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer Sentinel Mixture Models," *arXiv Prepr.*, Sep. 2016, Accessed: May 22, 2017. [Online]. Available: http://arxiv.org/abs/1609.07843.

[33] N. Kriegeskorte, "Representational similarity analysis – connecting the branches of systems neuroscience," *Front. Syst. Neurosci.*, vol. 2, 2008, doi: 10.3389/neuro.06.004.2008.

[34] I. A. Blank and E. Fedorenko, "Domain-general brain regions do not track linguistic input as closely as language-selective regions," *J. Neurosci.*, vol. 37, no. 41, pp. 9999–10011, Oct. 2017, doi: 10.1523/JNEUROSCI.3642-16.2017.

[35] E. Diachek, I. Blank, M. Siegelman, J. Affourtit, and E. Fedorenko, "The domain-general multiple demand (MD) network does not support core aspects of language comprehension: A large-scale fMRI investigation," *J. Neurosci.*, vol. 40, no. 23, pp. 4536–4550, Jun. 2020, doi: 10.1523/JNEUROSCI.2036-19.2020.

[36] C. Shain, I. A. Blank, M. van Schijndel, W. Schuler, and E. Fedorenko, "fMRI reveals language-specific predictive coding during naturalistic sentence comprehension," *Neuropsychologia*, vol. 138, Feb. 2020, doi: 10.1016/j.neuropsychologia.2019.107307.

[37] L. Wehbe *et al.*, "Incremental language comprehension difficulty predicts activity in the language network but not the multiple demand network," *bioRxiv Prepr.*, Apr. 2020, doi: 10.1101/2020.04.15.043844.

[38] K. Ethayarajh, "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings," in *Empirical Methods in Natural Language Processing (EMNLP)*, Sep. 2019, pp. 55–65, Accessed: Mar. 10, 2021. [Online]. Available: http://arxiv.org/abs/1909.00512.