Many but not all deep neural network audio models capture brain responses and exhibit hierarchical region correspondence

Greta Tuckute*[1,2], Jenelle Feather*[1,2], Dana Boebinger[1,2,3,4], Josh H. McDermott[1,2,3]
*co-first authors

[1]Department of Brain and Cognitive Sciences, McGovern Institute for Brain Research MIT, Cambridge, MA, USA
[2]Center for Brains, Minds, and Machines, MIT, Cambridge, MA, USA
[3]Program in Speech and Hearing Biosciences and Technology, Harvard, Cambridge, MA, USA
[4]University of Rochester Medical Center, Rochester, NY, USA

Abstract

Deep neural networks are commonly used as models of the visual system, but are less explored in audition. Prior work provided examples of audio-trained neural networks that produced good predictions of auditory cortical fMRI responses and exhibited correspondence between model stages and brain regions, but left it unclear whether these results generalize to other neural network models. We evaluated brain-model correspondence for publicly available audio neural network models along with in-house models trained on four different tasks. Most tested models out-predicted previous filter-bank models of auditory cortex, and exhibited systematic model-brain correspondence: middle stages best predicted primary auditory cortex while deep stages best predicted non-primary cortex. However, some state-of-the-art models produced substantially worse brain predictions. The training task influenced the prediction quality for specific cortical tuning properties, with best overall predictions resulting from models trained on multiple tasks. The results suggest the importance of task optimization in constraining brain representations.

Introduction

An overarching aim of neuroscience is to build quantitatively accurate computational models of sensory systems. Success entails models that take sensory signals as input and reproduce the behavioral judgments mediated by a sensory system as well as its internal representations. A model that can replicate behavior and brain responses for arbitrary stimuli would help validate the theories that underlie the model, but would also have a host of important applications. For instance, such models could guide brain-machine interfaces by specifying patterns of brain stimulation needed to elicit particular percepts or behavioral responses.

Advances in machine learning have enabled models that achieve human-level performance on real-world classification tasks such as object and speech recognition, stimulating interest in deep artificial neural networks (DNNs) as candidate models of vision, audition, language, and other domains[1–6]. Such models are relatively well explored within vision, where they reproduce some patterns of human behavior[7–10] and in many cases appear to replicate aspects of the hierarchical organization of the primate ventral stream[11–14]. These and other findings are consistent with the idea that brain representations are constrained by the demands of the tasks organisms must carry out, such that optimizing for ecologically relevant tasks produces better models of the brain.

Deep neural network models have also enabled progress in audition. Comparisons of human and model behavioral characteristics have found that audio-trained neural networks often reproduce patterns of human behavior when optimized for naturalistic tasks and stimulus sets[15–19]. Several studies have also compared audio-trained neural networks to brain responses within the auditory system[20,15,21–27].

The best-known of these prior studies is arguably that of Kell et al., (2018), who found that DNNs jointly optimized for speech and music classification could predict functional magnetic resonance imaging (fMRI) responses to natural sounds in auditory cortex substantially better than a standard model based on spectrotemporal filters. In addition, model stages exhibited correspondence with brain regions, with middle stages best predicting primary auditory cortex and deeper stages best predicting non-primary auditory cortex. Kell et al. measured brain predictions for a set of convolutional architectures, and found that models that performed better on the training task produced better brain predictions, as in other domains[28,11,29,30]. However, Kell et al. used only a fixed set of two tasks, investigated a single architecture class (feedforward convolutional architectures with no skip-layer connections), relied exclusively on regression-derived predictions of held-out responses as the metric of brain-model similarity, and only examined correspondence between model stages and brain regions for the best-performing model.

Several subsequent studies built on these findings by analyzing models trained in various ways to recognize and/or predict speech, and found they were able to predict cortical responses to speech better than chance, with some evidence that different model stages best predicted different brain regions[24,26,25,27]. But each of these studies also analyzed a small number of models, and each used a different brain data set, making it difficult to compare results across studies. It has thus remained unclear whether DNNs trained on other tasks also produce good predictions of brain responses, whether the correspondence between model stages and brain regions is

consistent across models, whether the findings generalize to sounds other than speech, and whether the training task critically influences the ability to predict responses in particular parts of auditory cortex. These questions are important for substantiating the hierarchical organization of the auditory cortex, for understanding the role of tasks in shaping cortical representations, and for guiding the development of better models of the auditory system.

Here we examine brain-DNN similarities within the auditory cortex for a large set of models. The primary similarity metric we assume is the variance explained by linear mappings fit from model features to brain responses. The variance explained by linear mappings has the advantage of being in widespread use and hence facilitates comparison of results to related work. To ensure that the conclusions were robust to the choice of metric, we repeated key analyses using representational similarity metrics[31].

Our goal was to probe the generality of brain-DNN similarities in the auditory cortex as well as their dependence on the training task. To address generality, we tested a large set of publicly available audio-trained neural network models, trained on a wide variety of tasks and spanning a large range of model architectures. To specifically address the effect of training task, we supplemented these publicly available models with in-house models trained on four different tasks. We evaluated both the overall quality of the brain predictions as compared to a standard baseline spectrotemporal filter model of the auditory cortex (Chi et al., 2005), as well as the correspondence between model stages and brain regions. We used two different fMRI data sets to assess the reproducibility and robustness of the results: the original data set (Norman-Haignere et al., 2015, n=8) used in Kell et al., to facilitate comparisons to those earlier results, as well as a second recent data set (Boebinger et al., 2021, n=20) with data from a total of 28 unique participants.

We found that most deep neural network models produced better predictions of brain responses than the baseline model of the auditory cortex. In addition, most models exhibited a correspondence between model stages and brain regions, with lateral, anterior, and posterior non-primary auditory cortex being better predicted by deeper model stages. However, not all models produced good predictions, suggesting that some training tasks and architectures yield more brain-like predictions than others. We also observed significant effects of the training task on the predictions of speech, music, and pitch-related cortical responses. The best overall predictions were produced by models trained on multiple tasks. The results indicate that many deep neural networks replicate aspects of auditory cortical computation, but indicate the important role of training tasks in obtaining models that yield accurate brain predictions.

Results

Model selection

We began by compiling a set of models that we could compare to brain data (see “Candidate models” in Methods for full details and Tables 1 and 2 for an overview). Two criteria dictated the choice of models. First, we sought to survey a range of models encompassing a wide set of architectures and optimization criteria, to assess the generality with which deep neural networks would be able to model auditory cortical responses. Second, we wanted to explore effects of the

training task. The main constraint on the model set was that there were relatively few publicly available audio-trained deep neural network models available at the time of this study (in part because much work on audio engineering is done in industry settings where models and data sets are not made public). We thus included every model we could obtain in a PyTorch implementation that had been trained on some sort of audio task at a realistic scale (i.e., we neglected models trained to classify spoken digits, or other small-scale tasks, on the grounds that such tasks are unlikely to place strong constraints on the model representations). The resulting set of 9 models varied in both their architecture (spanning convolutional neural networks, recurrent neural networks, and transformers) and training task (ranging from automatic speech recognition and speech enhancement to audio captioning and audio source separation).

To supplement these external models, we trained ten models ourselves: two architectures trained separately on each of four tasks as well as on three of the tasks simultaneously. We used the three tasks that could be implemented using the same data set (where each sound clip had labels for words, speakers, and environmental sounds). One of the architectures we used was similar to that used in our earlier study (Kell et al., 2018), which identified a candidate architecture from a large search over number of stages, location of pooling, and size of convolutional filters. The resulting model performed well on both word and music genre recognition, and was predictive of brain responses to natural sounds. This architecture (henceforth CochCNN9) consisted of a sequence of convolutional, normalization, and pooling stages preceded by a hand-designed model of the cochlea (henceforth termed a 'cochleagram'). The second architecture was a ResNet50[35] backbone with a cochleagram front end (henceforth CochResNet50). CochResNet50 was a much deeper model than CochCNN9 (50 layers compared to 9 layers) with residual (skip-layer) connections, and although this architecture was not determined via an explicit architecture search for auditory tasks, it had been previously optimized for other perceptual tasks, and outperformed CochCNN9 on the training tasks (see Methods; Candidate models). We used two architectures to obtain a sense of the consistency of any effects of task that we might observe.

The four in-house training tasks consisted of recognizing words, speakers, environmental sounds, and musical genres from audio (referred to henceforth as Word, Speaker, AudioSet and Genre, respectively). The multi-task models had three different output layers, one for each included task (word, speaker, and environmental sound recognition), connected to the same network. The three tasks for the multi-task network were originally chosen because we could train on all of them simultaneously using a single existing data set (the Word-Speaker-Noise data set[36]) in which each clip has three associated labels: a word, a speaker, and an environmental sound (sounds from the AudioSet data set[37]) background. For the single-task networks, we used one of these three sets of labels. We additionally trained models with a fourth task – a music-genre classification task originally presented in Kell et al., (2018). As it turned out, the first three tasks individually produced better brain predictions than the fourth, and the multi-task model produced better predictions than any of the models individually, and so we did not explore additional combinations of tasks. These in-house models were intended to allow a controlled analysis of the effect of task, to complement the all-inclusive but uncontrolled set of external models.

We compared each of these models to an untrained baseline model that is commonly used in cognitive neuroscience[32]. The baseline model consisted of a set of spectrotemporal filters applied to a model of the cochlea (henceforth referred to as the SpectoTemporal model).

Brain data

To assess the replicability and robustness of the results, we evaluated the models on two independent fMRI data sets (each with three scanning sessions per participant). Each presented the same set of 165 two-second natural sounds to human listeners. One experiment[33] collected data from 8 participants with moderate amounts of musical experience (henceforth NH2015). This data set was analyzed in a previous study investigating deep neural network predictions of fMRI responses[15]. The second experiment[34] collected data from a different set of 20 participants, 10 of whom had almost no musical experience, and 10 of whom had extensive musical training (henceforth B2021). The fMRI experiments measured the blood-oxygen-level-dependent (BOLD) response to each sound in each voxel in the auditory cortex of each participant (including all temporal lobe voxels that responded significantly more to sound than silence, and whose test-retest response reliability exceeded a criterion; see Methods; fMRI data).

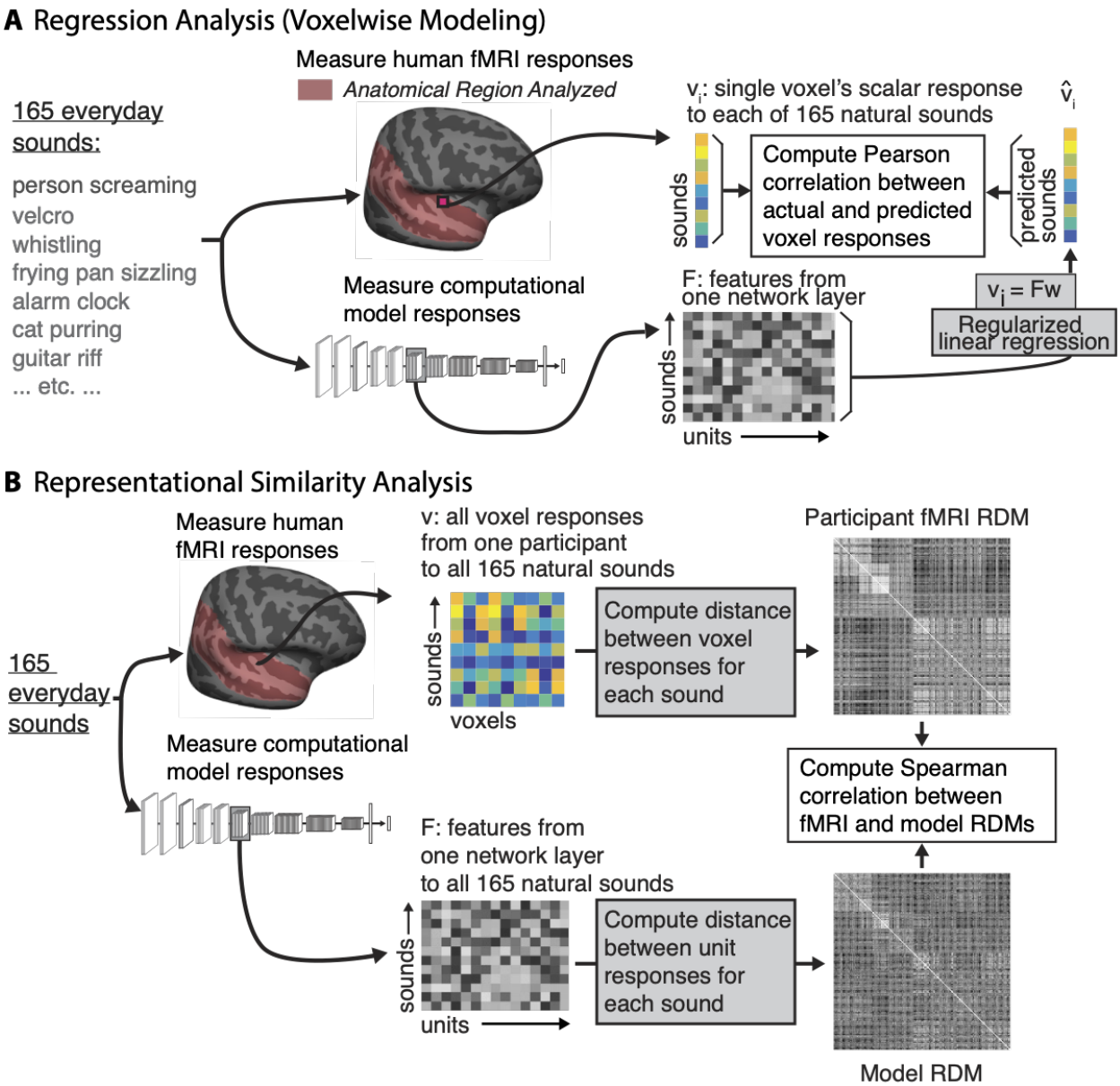


Figure 1. Analysis method. (A) Regression Analysis (voxelwise modeling). Brain activity of human participants (n=8, n=20) was recorded while they listened to a set of 165 natural sounds in the fMRI scanner. Data were taken from two previous publications[33,34]. We then presented the same set of 165 sounds to each model, measuring the time-averaged unit activations from each model stage in response to each sound. We performed an encoding analysis where voxel activity was predicted by a regularized linear model of the DNN activity. We modeled each voxel as a linear combination of model units from a given model stage, estimating the linear transform with half (n=83) the sounds and measuring the prediction quality by correlating the empirical and predicted response to the left-out sounds (n=82) using the Pearson correlation. We performed this procedure for 10 random splits of the sounds. Figure adapted from Kell et al.[15]. **(B)** Representational Similarity Analysis (RSA). We used the set of brain data and model activations described for the voxelwise regression modeling. We constructed a representational dissimilarity matrix (RDM) from the fMRI responses by computing the distance (1-Pearson correlation) between all voxel responses to each pair of sounds. We similarly constructed an RDM from the unit responses from a model stage to each pair of sounds. We measured the Spearman correlation between the fMRI and model RDMs as the metric of

brain-model similarity. When reporting this correlation from a best model stage, we used 10 random splits of sounds, choosing the best stage from the training set of 83 sounds and measuring the spearman correlation for the remaining set of 82 test sounds. The fMRI RDM is the average RDM across all participants for all voxels and all sounds in NH2015. The model RDM is from an example model stage (ResNetBlock_2 of the CochResNet50-multitask network).

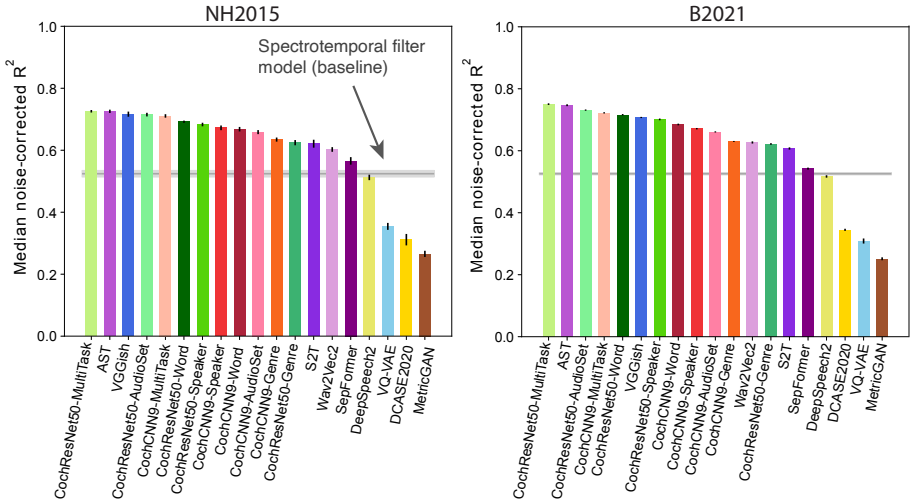
General approach to analysis

Because the sounds were short relative to the time constant of the fMRI BOLD signal, we summarized the fMRI response as a single scalar value for each sound. As shown in Figure 1, each sound was passed through a neural network model, and the unit activations from each network stage were used to predict the response of individual voxels (after averaging unit activations over time to mimic the slow time constant of the BOLD signal). Predictions were generated with cross-validated ridge regression, using methods similar to those of many previous studies using encoding models of fMRI measurements[38–41,15,42]. The specific approach here was modeled after that of Kell et al., 2018: we used 83 of the sounds to fit the linear mapping from model units to a voxel's response, and then evaluated the predictions on the 82 remaining sounds, taking the median across 10 training/test cross-validation splits, and correcting for both the reliability of the measured voxel response and the reliability of the predicted voxel response [43,44]. The variance explained by a model stage was taken as a metric of the brain-likeness of the model representations. We asked i) to what extent the models in our set were able to predict brain data, and ii) whether there was a relationship between stages in a model and regions in the human brain.

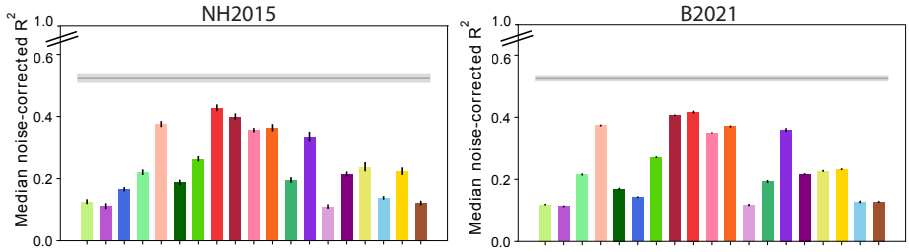
To assess the robustness of our conclusions to the evaluation metric, we also compared a measure of representational similarity calculated for brain and model responses. We measured representational dissimilarity matrices (RDMs) for a set of voxel responses from the Pearson correlation of all the voxel responses to one sound with that for another sound (Figure 1B). These correlations for all pairs of sounds yields a matrix, which is standardly expressed as 1-C, where C is the correlation matrix. When computed from all voxels in the auditory cortex, this matrix is highly structured, with some pairs of sounds producing much more similar responses than others (Supplementary Figure S1). We analogously measured this matrix from the time-averaged unit responses within a model stage, and then evaluated the similarity of the brain and model RDMs via their Spearman correlation. As in previous work[28,45], we did not correct this metric for the reliability of the RDMs, but instead computed a noise ceiling for it. We estimated the noise ceiling as the correlation between a held-out participant's RDM and the average RDM of the remaining participants.

A Regression

i Trained Networks

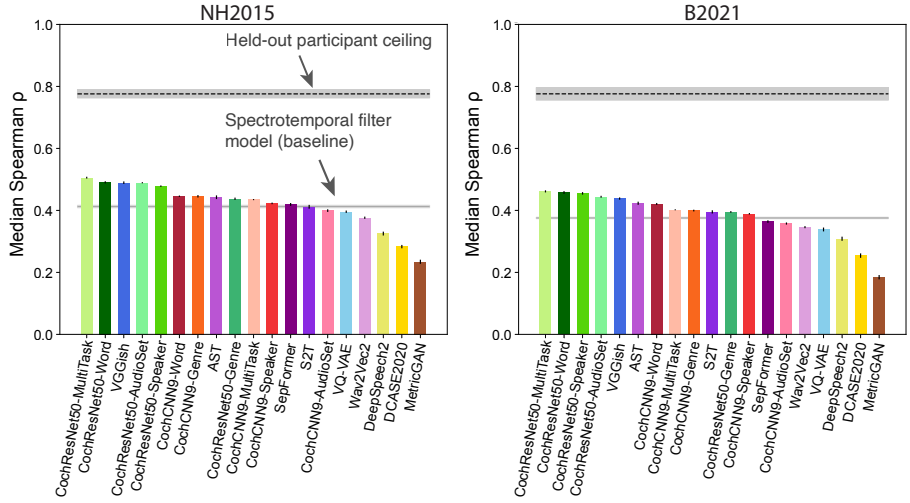


ii Permuted Networks



B Representational Similarity

i Trained Networks



ii Permuted Networks

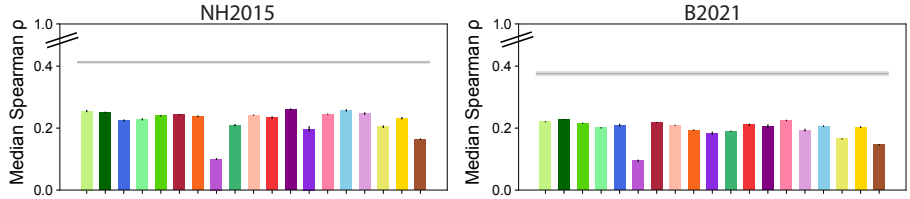


Figure 2. Evaluation of overall model-brain similarity. (A) Using regression, explained variance was measured for each voxel and the aggregated median variance explained was obtained for the best-predicting stage for each model, selected using independent data. Grey line shows variance explained by the SpectroTemporal baseline model. Colors indicate the nature of the model architecture: CochCNN9 architectures in shades of red, CochResNet50 architectures in shades of green, Transformer architectures in shades of violet (AST, Wav2vec2, S2T, SepFormer), recurrent architectures in shades of yellow (DCASE2020, DeepSpeech2), other convolutional architectures in shades of blue (VGGish, VQ-VAE), and miscellaneous in brown (MetricGAN). Panel **i** shows the trained networks, and panel **ii** shows the control networks with permuted weights. Error bars are within-participant SEM. Note that error bars are smaller for the B2021 data set because of the larger number of participants (n=20 vs. n=8). For both data sets, most trained models out-predict the baseline model, whereas all permuted models produce worse predictions than the baseline. **(B)** We analyzed the representational similarity between all auditory cortex fMRI responses and the computational models. The models and colors are the same as in (A). The dashed black line shows the noise ceiling measured by comparing one participant's RDM with the average of the RDMs from each of the other participants. Panel **i** shows the trained networks, and panel **ii** shows the control networks with permuted weights. Error bars are within-participant SEM. Similar to the regression analysis, many of the trained models exhibit RDMs that are more correlated with the human RDM than the baseline model, whereas all permuted models are less correlated than the baseline.

Many DNN models outperform traditional models of the auditory cortex

We first assessed the overall quality of the brain predictions for each model using regularized regression, aggregating across all voxels in the auditory cortex. For each model, explained variance was measured for each voxel using the best-predicting stage for that voxel (see Methods; Voxel response modeling). We then took the median of this explained variance across voxels for each model (averaged across participants).

As shown in Figure 2Ai, the best-predicting stage of most trained DNN models produced better overall predictions of auditory cortex responses than did the standard SpectroTemporal baseline model[32] (see Supplementary Figure S2 for predictivity across model stages). This was true for all of the in-house models as well as about half of the external models developed in engineering contexts. However, some models developed in engineering contexts did not produce good predictions, substantially under-predicting the baseline model. The heterogeneous set of external models was intended to provide a strong test of the generality of brain-DNN relations, and sacrificed controlled comparisons between models (because models differed on many dimensions). It is thus difficult to pinpoint the factors that cause some models to produce poor predictions. This finding nonetheless demonstrates that some models that are trained on large amounts of data, and that perform some auditory tasks well, do not accurately predict auditory cortical responses. But the results also show that many models produce better predictions than the classical SpectroTemporal baseline model.

Brain predictions of DNN models depend critically on task-optimization

To assess whether the improved predictions compared to the SpectroTemporal baseline model could be entirely explained by the DNN architectures, we performed the same analysis with each model's parameters (e.g., weights, biases) permuted within each model stage (Figure 2Aii). This model manipulation destroyed the parameter structure learned during task-optimization, while preserving the model architecture and the marginal statistics of the model parameters. This was

intended as replacement for testing untrained models with randomly initialized weights[15], the advantage being that it seemed a more conservative test for the external models, for which the initial weight distributions were in some cases unknown.

In all cases, these control models produced worse predictions than the trained models, and in no case did they out-predict the baseline model. This result indicates that task optimization is consistently critical to obtaining good brain predictions. This conclusion is consistent with previously published results[15] but substantiates them on a much broader set of models and tasks.

Qualitatively similar conclusions from representational similarity

To ensure that the conclusions from the regression-based analyses were robust to the choice of metric, we repeated our analyses using representational similarity. Analyses of representational similarity gave qualitatively similar results to those with regression. We computed the Spearman correlation between the RDM for all auditory cortex voxels and that for the unit activations of each stage of each model, choosing the model stage that yielded the best match. We used 83 of the sounds to choose the best-matching model stage, and then measured the model-brain RDM correlation for RDMs computed for the remaining 82 sounds. We performed this procedure with 10 different splits of the sounds, averaging the correlation across the 10 splits. This analysis showed that most of the models in our set had RDMs that were more correlated with the human auditory cortex RDM than that of the baseline model (Figure 2Bi). Moreover, the two measures of brain-model similarity (variance explained and correlation of RDMs) were correlated in the trained networks (r^2=0.75 for NH2015 and r^2=0.78 for B2021, $p<<.001$), with models that showed poor matches on one metric tending to show poor matches on the other. The correlations with the human RDM were nonetheless well below the noise ceiling, indicating that none of the models fully accounts for the fMRI representational similarity. As expected, the RDMs for the permuted models were less similar to that for human auditory cortex, never exceeding the correlation of the baseline model (Figure 2Bii). Overall, these results provide converging evidence for the conclusions of the regression-based analyses.

Improved predictions of DNN models are most pronounced for pitch, speech, and music-selective responses

To examine the model predictions for specific tuning properties of the auditory cortex, we used a previously derived set of cortical response components. Previous work[33] found that cortical voxel responses to natural sounds can be explained as a linear combination of six response components (Figure 3A). These six components can be interpreted as capturing the tuning properties of underlying neural populations. Two of these components were well accounted for by audio frequency tuning, and two others were relatively well explained by tuning to spectral and temporal modulation frequencies. One of these latter two components was selective for sounds with salient pitch. The remaining two components were highly selective for speech and music, respectively. The six components had distinct (though overlapping) anatomical distributions, with the components selective for pitch, speech, and music most prominent in different regions of non-primary auditory cortex (Figure 3B; components 4-6, selective for pitch, speech, and music, respectively). These components provide one way to examine whether the improved model predictions seen in Figure 2 are specific to particular aspects of cortical tuning.

We again used regression to generate model predictions, but this time using the component responses rather than voxel responses (Figure 3C). We fit a linear mapping from the unit activations in a model stage (for a subset of “training” sounds) to the component response, then measured the predictions for left-out “test” sounds, averaging the predictions across test splits. The main difference between the voxel analyses and the component analyses is that we did not noise-correct the estimates of explained component variance. This is because we could not estimate test-retest reliability of the components, as they were derived with all three scans worth of data.

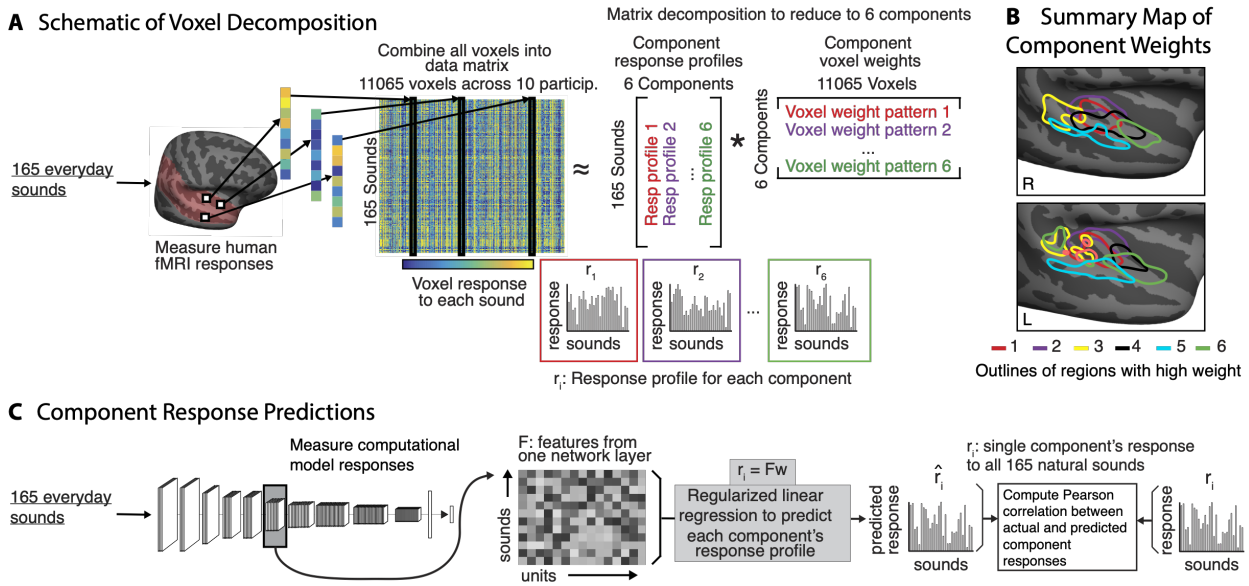


Figure 3. **Component decomposition of fMRI responses. (A)** Voxel component decomposition method. The voxel responses of a set of participants are approximated as a linear combination of a small number of component response profiles. The solution to the resulting matrix factorization problem is constrained to maximize a measure of the non-Gaussianity of the component weights. Voxel responses in auditory cortex to natural sounds are well accounted for by six components. Figure adapted from a previous publication (Norman-Haignere et al., 2015). **(B)** The six components are concentrated in different regions of the auditory cortex. Figure adapted from a previous publication (Norman-Haignere et al., 2015). **(C)** We generated model predictions for each component’s response using the same approach used for voxel responses, in which the model unit responses were combined to best predict the component response, with explained variance measured in held-out sounds (taking the median of the explained variance values obtained across train/test cross-validation splits).

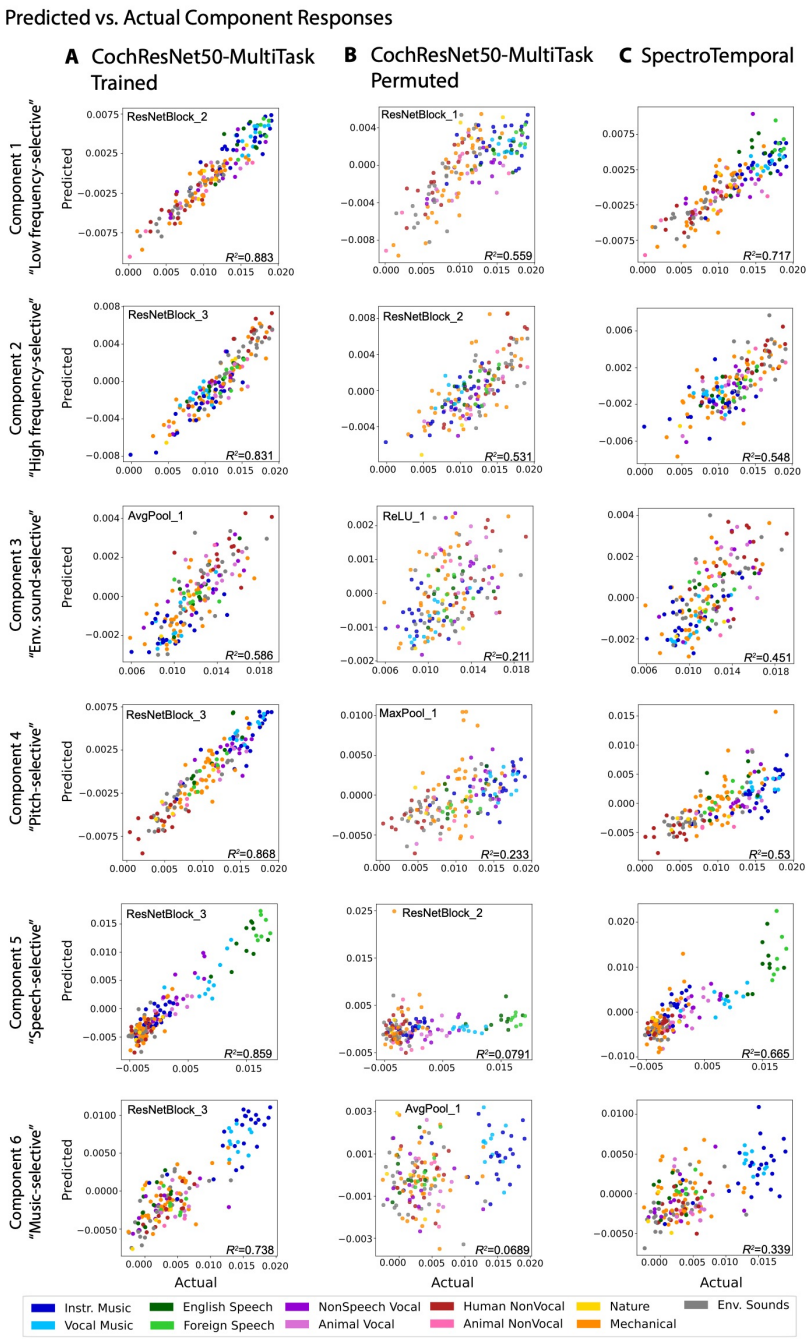


Figure 4. Example model predictions for six components of fMRI responses to natural sounds. (A) Predictions of the six components by a trained deep neural network model (CochResNet50-MultiTask). Each dot corresponds to a single natural sound from the set of 165. Dot color denotes the sound's semantic category. Model predictions were made from the model stage that best predicted a component's response. The predicted response is the average of the predictions for a sound across the test half of 10 different train-test splits (including each of the splits for which the sound was present in the test half). **(B)** Predictions of the six components by same model used in (A) but with permuted weights. Predictions are substantially worse than for the trained model, indicating that task optimization is important for obtaining good predictions, especially for components 4-6. **(C)** Predictions of the six components by the SpectroTemporal model. Predictions are substantially worse than for the trained model, particularly for components 4-6.

Figure 4A shows the actual component responses (from the Norman-Haignere et al., 2015 data set) plotted against the predicted responses for the best-predicting model stage (selected separately for each component) of the multi-task CochResNet50, which gave the best overall voxel response predictions (Figure 2). The model replicates most of the variance in all components (between 58% and 88% of the variance, depending on the component). Given that two of the components are highly selective for particular categories, one might suppose that the good predictions in these cases could be primarily due to predicting higher responses for some categories than others, and the model indeed reproduces the differences in responses to different sound categories (e.g. with high responses to speech in the speech-selective component, and high responses to music in the music-selective component). However, it also replicates some of the response variance within sound categories. For instance, the model predictions explained 51.7% of the variance in responses to speech sounds in the speech-selective component, and 55.1% of the variance in the responses to music sounds in the music-selective component (both of these values are much higher than would be expected by chance; speech: $p=.001$; music: $p<.001$). We note that although we could not estimate the reliability of the components in a way that could be used for noise correction, we were able to measure their similarity between different groups of participants, and this was lowest for component 3, followed by component 6[34]. Thus, the differences between components in the overall quality of the model predictions are plausibly related to their reliability.

The component response predictions were much worse for models with permuted weights, as expected given the results of Figure 2 (Figure 4B; results shown for the permuted multi-task CochResNet50; results were similar for other models with permuted weights). The notable exceptions were the first two components, which reflect frequency tuning[33]. This is likely because frequency information is made explicit by a convolutional architecture operating on a cochlear representation, irrespective of the model weights. For comparison we also show the component predictions for the SpectroTemporal baseline model (Figure 4C). These are better than those of the permuted model (one-tailed $p<.001$; permutation test), but significantly worse than those of the best-predicting trained model for all six components (one-tailed $p<<.0001$; permutation test).

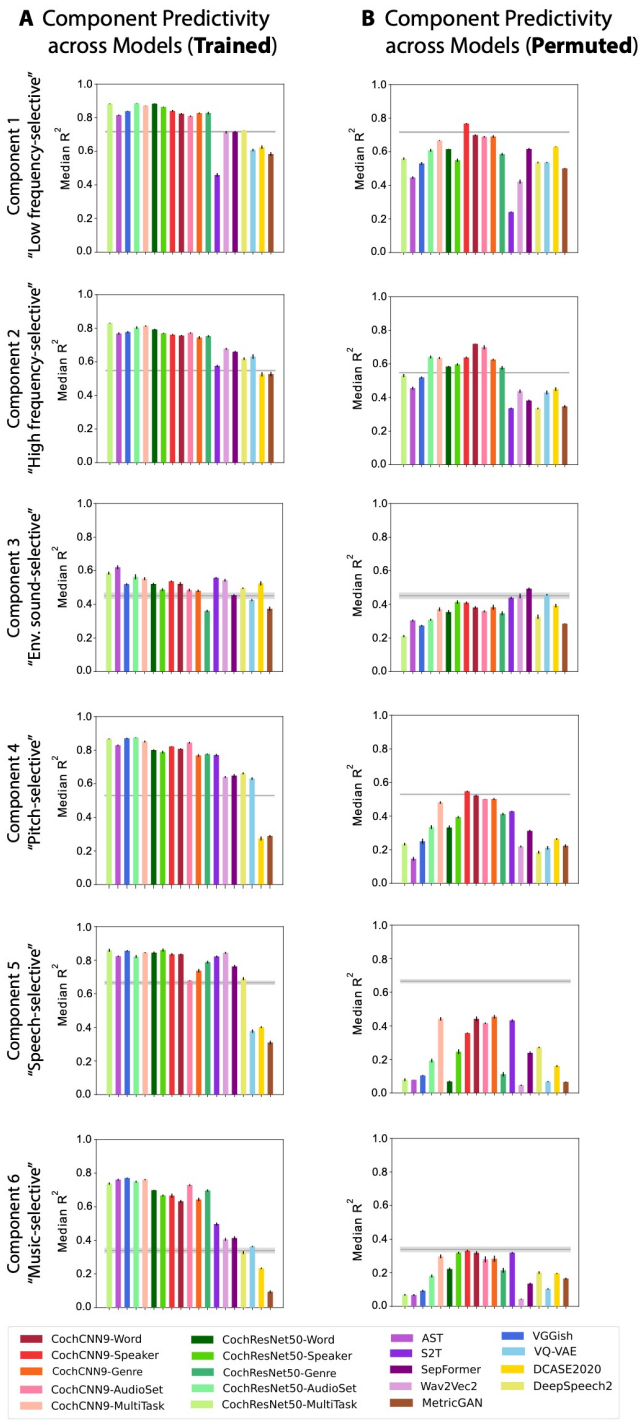


Figure 5. Summary of model predictions of fMRI response components. (A) Component response variance explained by each of the trained models. Model ordering is the same as that in Figure 2A for ease of comparison. Variance explained was obtained from the best-predicting stage of each model for each component. Error bars plot SEM over iterations of the model stage selection procedure (see Methods; Component modeling). **(B)** Component response variation explained by each of the permuted models. The trained models, but not the permuted models, tend to out-predict the SpectroTemporal baseline for all components, but the effect is most pronounced for components 4-6.

These findings held across most of the neural network models we tested. Most of the trained neural network models produced better predictions than the SpectroTemporal baseline model for most of the components (Figure 5A), with the improvement being specific to the trained models (Figure 5B). However, it is also apparent that the difference between the trained and permuted models is most pronounced for components 4-6 (selective for pitch, speech, and music, respectively; compare Figure 5A to 5B). This result indicates that the improved predictions for task-optimized models are most pronounced for higher-order tuning properties of auditory cortex.

Many DNN models exhibit model-stage-brain-region correspondence with auditory cortex

One of the most intriguing findings from the neuroscience literature on deep neural network models is that the models often exhibit a correspondence with the hierarchical organization of sensory systems[11–15,46]. To explore the generality of this correspondence for audio-trained models, we examined the best-predicting model stage for each voxel of each participant in the two fMRI data sets, separately for each model.

We first plotted the best-predicting stage as a surface map displayed on an inflated brain. The best-predicting model stage for each voxel was expressed as a number between 0 and 1, and we plot the median of this value across participants. In both data sets, earlier model stages tended to produce the best predictions of primary auditory cortex while deeper model stages produced better predictions of non-primary auditory cortex. We show these maps for the eight best-predicting models in Figure 6A, and provide them for all remaining models in Supplementary Figure S3. There is some variation from model to model, both in the relative stages that yield the best predictions, and in the detailed anatomical layout of the resulting maps, but the differences between primary and non-primary auditory cortex are fairly consistent across models. The stage-region correspondence was specific to the trained models; the models with permuted weights produce relatively uniform maps (Supplementary Figure S4).

To summarize these maps across models, we computed the median best-stage for each voxel across all 15 models that produced better overall predictions compared to the baseline model (Figure 2A). This average best-stage map (Figure 6B) shows a clear gradient, with voxels in and around primary auditory cortex (black outline) best predicted by earlier stages than voxels beyond primary auditory cortex. This correspondence is lost when the weights are permuted (Figure 6C).

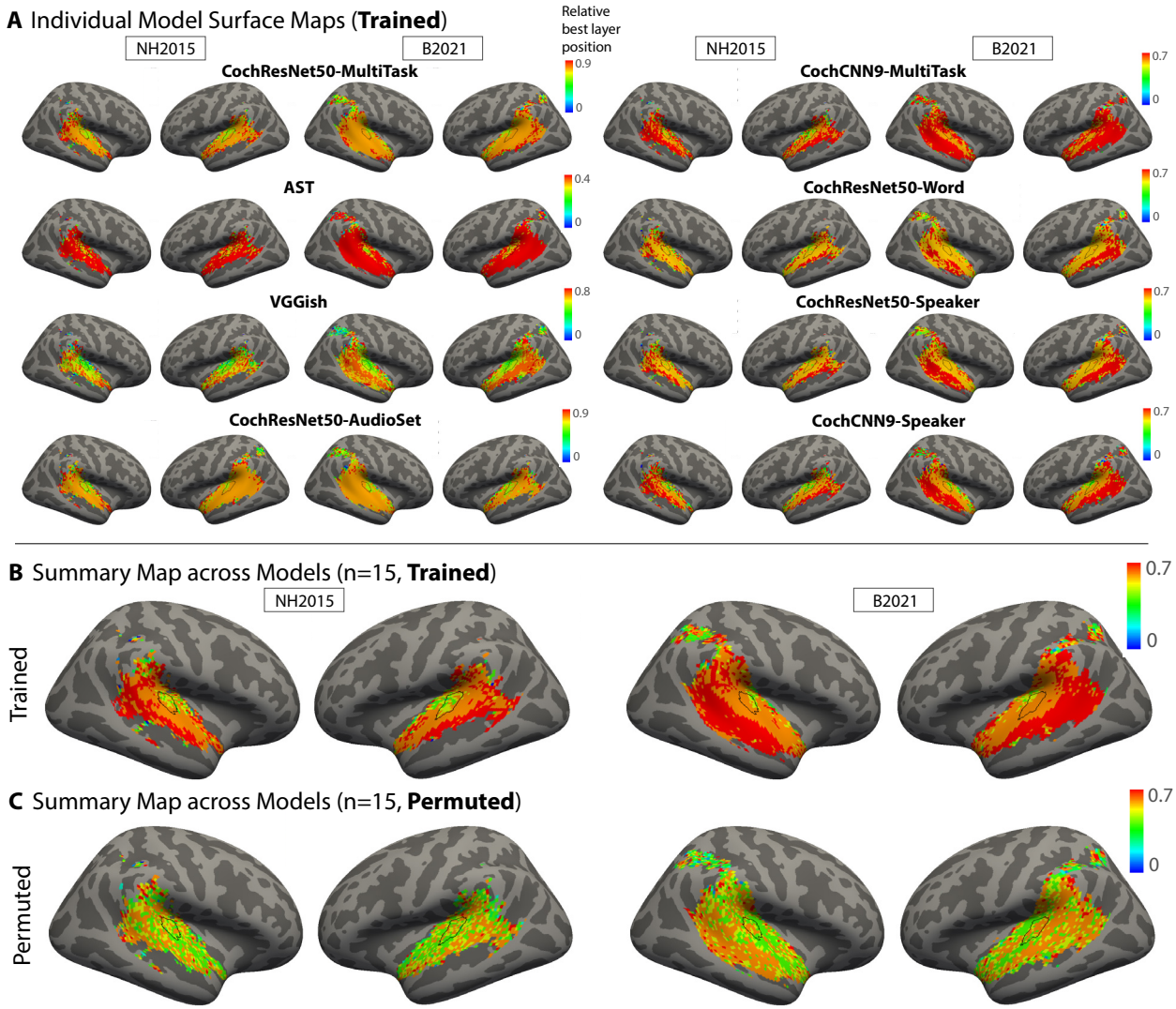


Figure 6. **Surface maps of best-predicting model stage**. **(A)** To investigate correspondence between model stages and brain regions, we plot the model stage that best predicts each voxel as a surface map (FsAverage) (median best stage across participants). We assigned each model stage a position index between 0 and 1 (using minmax normalization such that the first stage is assigned a value of 0 and the last stage a value of 1). We show this map for the eight best-predicting models as evaluated by the median noise-corrected R^2 plotted in Figure 2A (see Supplementary Figure S3 for maps from other models). The color scale limits were set to extend from 0 to the stage beyond the most common best stage (across voxels). We found that setting the limits in this way made the variation across voxels in the best stage visible by not wasting dynamic range on the deep model stages, which were almost never the best-predicting stage. For both data sets, middle stages best predict primary auditory cortex, while deep stages best predict non-primary cortex. **(B)** Best-stage map averaged across all models that produced better predictions than the baseline SpectroTemporal model. The map plots the median value across models, and thus is composed of discrete color values. The thin black outline plots the borders of an anatomical ROI corresponding to primary auditory cortex. **(C)** Best-stage map for the same models as in (B), but with permuted weights.

To quantify the trends that were evident in the surface maps, we computed the average best stage within four anatomical regions of interest (ROIs): one for primary auditory cortex, along with

three ROIs for posterior, lateral, and anterior non-primary auditory cortex. These ROIs were combinations of subsets of ROIs in the Glasser et al. parcellation[47] (Figure 7A). We note that they were taken directly from a previous publication[34], where they were intended to capture the auditory cortical regions exhibiting reliable responses to natural sounds, and were not adapted in any way to the present analysis. We visualize the results of this analysis by plotting the average best stage for the primary ROI vs. that of each of the non-primary ROIs, expressing the stage's position within each model as a number between 0 and 1 (Figure 7B). In each case, nearly all models lie above the diagonal (Figure 7C), indicating that all three regions of non-primary auditory cortex are consistently better predicted by deeper model stages compared to primary auditory cortex, irrespective of the model. This result was highly statistically significant in each case (Wilcoxon signed rank test: two-tailed $p<.005$ for all six comparisons; two data sets x three non-primary ROIs).

To confirm that these results were not merely the result of the deep neural network architectural structure (for instance, with pooling operations tending to produce larger receptive fields at deeper stages compared to earlier stages), we performed the same analysis on the models with permuted weights. In this case the results showed no evidence for a mapping between model stages and brain regions (Supplementary Figure S5, right; no significant difference between primary and non-primary ROIs in any of the six cases; Wilcoxon signed rank tests, two-tailed $p>0.33$ in all cases). This result is consistent with the surface maps (Figure 6C and Supplementary Figure S4), which tended to be fairly uniform.

We repeated the ROI analysis using representational similarity to determine the best-matching model stage for each ROI, and obtained similar results. The model stages that were most similar to the non-primary ROIs were again situated later than the model stage most similar to the primary ROI, in both data sets (Figure 7D; Wilcoxon signed rank test: two-tailed $p<.007$ for all six comparisons; two data sets x three non-primary ROIs). The model stages that provided the best match to each ROI according to each of the two metrics (regression and representational similarity) were correlated ($r^2=0.27$ for NH2015 and $r^2=0.24$ for B2021, measured across the 60 best stage values from 15 trained models for the four ROIs of interest, $p<<.001$ in both cases). This correlation is highly statistically significant but is nonetheless well below the maximum it could be given the reliability of the best stages (conservatively estimated as the correlation of the best stage between the two fMRI data sets; $r^2=0.91$ for regression and $r^2=0.94$ for representational similarity). This result suggests that the two metrics capture different aspects of brain-model similarity and that they do not fully align for the models we have at present, even though the general trend for deeper stages to better predict non-primary responses is present in both cases. Another difference between the results from the two metrics was that when evaluated via representational similarity, there was a trend for some of the non-primary ROIs to have later best-stages than the primary ROI for the permuted models as well ($p<.05$ in 3 of the 6 possible comparisons). However, these effects were weaker than those for the trained models (Cohen's $d<0.31$ for all permuted model comparisons compared to Cohen's $d>0.58$ for all trained model comparisons), indicating that the result was again not purely a function of the model architecture.

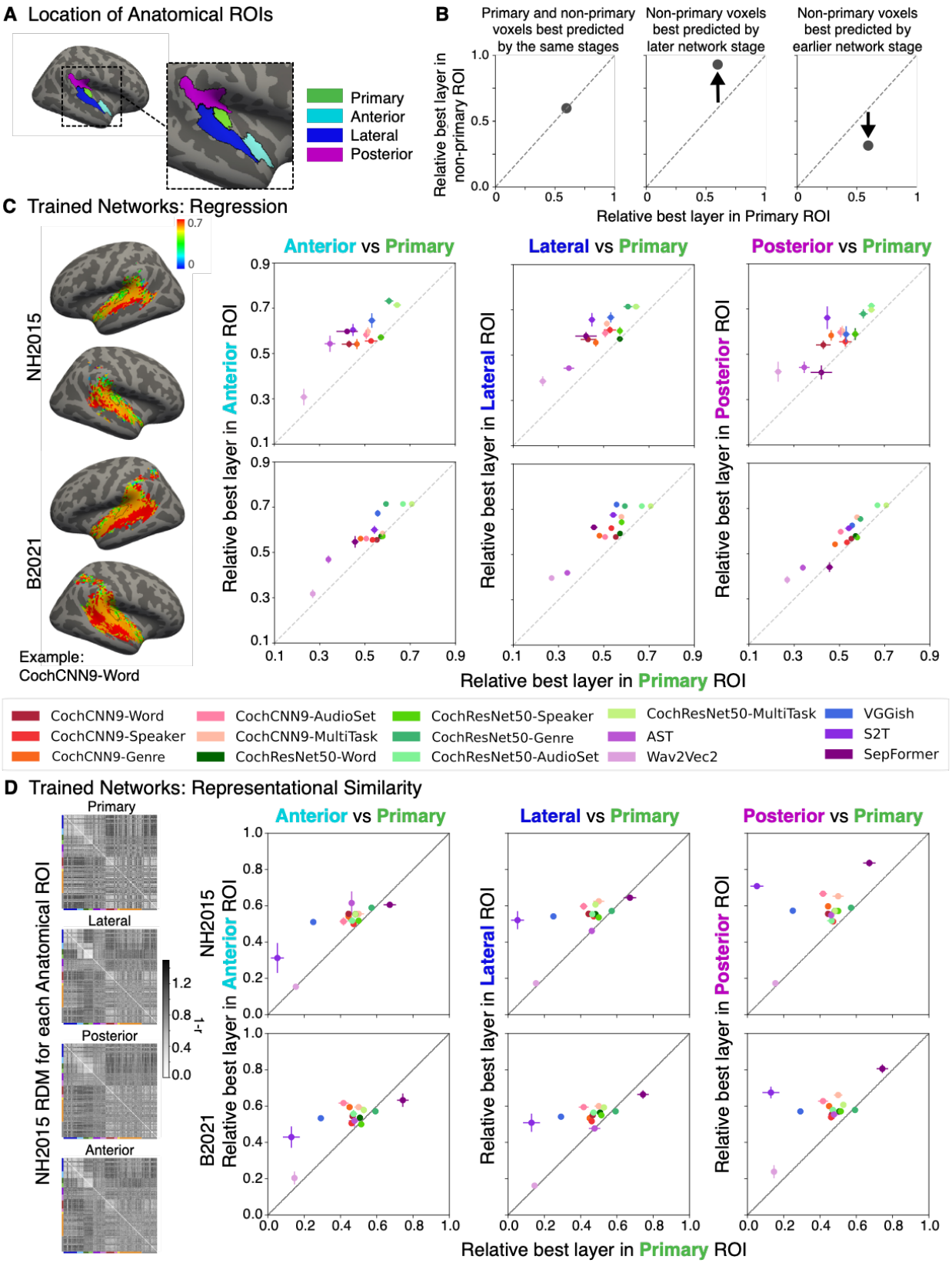


Figure 7. Nearly all DNN models exhibit stage-region correspondence. (A) Anatomical ROIs for analysis. ROIs were reproduced from a previous study[34], in which they were derived by pooling ROIs from

the Glasser anatomical parcellation[47]. **(B)** To summarize the model-stage-brain-region correspondence across models, we obtained the median best-predicting stage for each model within the four anatomical ROIs from A: primary auditory cortex (x axis in each plot in C and D) and anterior, lateral, and posterior non-primary regions (y axes in C and D). **(C)** We performed the analysis on each of the two fMRI data sets, including each model that out-predicted the baseline model in Figure 2 (n=15 models). Each data point corresponds to a model, with the same color correspondence as in Figure 2. Error bars are within-participant SEM. The non-primary ROIs are consistently best-predicted by later stages than the primary ROI. **(D)** Same analysis as (C) but with the best-matching model stage determined by correlations between the model and ROI representational dissimilarity matrices. RDMs for each anatomical ROI (left) are grouped by sound category, indicated by colors on the left and bottom edges of each RDM (same color-category correspondence as in Figure 4). Larger-scale fMRI RDMs for each ROI including the name of each sound is provided in Supplemental Figure S1.

Overall, these results are consistent with the hierarchical stage-region findings of Kell et al. (2018), but show that they apply fairly generally to a wide range of DNN models, that they replicate across different brain data sets, and are generally consistent across different analysis methods. The results suggest that the different representations learned by early and late stages of DNN models map onto differences between primary and non-primary auditory cortex in a way that is fairly consistent across a diverse set of models. This finding provides additional evidence for hierarchical computation in the auditory system.

Training task modulates model predictions

Many models produced good predictions of auditory cortical brain responses, in that they out-predicted the SpectroTemporal baseline model (Figure 2). But some models gave better predictions than others, raising the question of what causes differences in model predictions. To address this question, we analyzed the brain predictions of the in-house models, which consisted of the same two architectures trained on different tasks.

The results shown in Figure 2 indicate that some of our in-house tasks produced better overall predictions than others, and that the best overall model as evaluated with either metric (regression or RDM similarity) was that trained on three of the tasks (the CochResNet50-MultiTask). We note that the training stimuli were the same for the first three in-house tasks as for the multi-task model, with the only difference being the labels that were used to compute the training loss. Thus, any differences indeed reflect the task rather than the training stimuli. To gain insight into the source of these effects, we examined the in-house model predictions for the six components of auditory cortical responses (Figure 3) that vary across brain regions. The components seemed a logical choice for an analysis of the effect of task on model predictions because they isolate distinct cortical tuning properties. We focused on the pitch-selective, speech-selective, and music-selective components, because these showed the largest effects of model training (components 4-6, Figure 4&5), and because the tasks that we trained on seemed a priori most likely to influence representations of these types of sounds. This analysis was necessarily restricted to the regression-based model predictions because RDMs are not defined for any single component's response.

Comparisons of the variance explained in each component revealed interpretable effects of the training task (Figure 8). The pitch-selective component was best-predicted by the models trained on environmental sound recognition (R^2 was higher for AudioSet than for the other three tasks in both the CochCNN9 and CochResNet50 architectures, one-tailed $p<.0005$ for all six comparisons, permutation test). The speech-selective component was best predicted by the models trained on speech tasks. This was true both for the word recognition task (R^2 higher for the word-trained model than for the genre or AudioSet-trained models for both architectures, one-tailed $p<.05$ for all four comparisons) and for the speaker recognition task (one-tailed $p<.005$ for all four comparisons). Finally, the music-selective component was best-predicted by the models trained on environmental sound recognition (R^2 higher for the AudioSet-trained model than for the word-, speaker- or genre-trained models for both architectures, $p<<.0001$ for all six comparisons). We note that the AudioSet task contains multiple music classes, which plausibly explains its success in predicting this component. We note also that the component was less well predicted by the models trained to classify musical genre. This latter result may indicate that the genre data set/task does not fully tap into the features of music that drive cortical responses.

The differences between tasks were most evident in scatter plots of the variance explained for pairs of components (Figure 8B). For instance, the speech-trained models are furthest from the diagonal when the variance explained in the speech and music components are compared. And the AudioSet-trained models, along with the multi-task models, are well separated from the other models when the pitch- and music-selective components are compared.

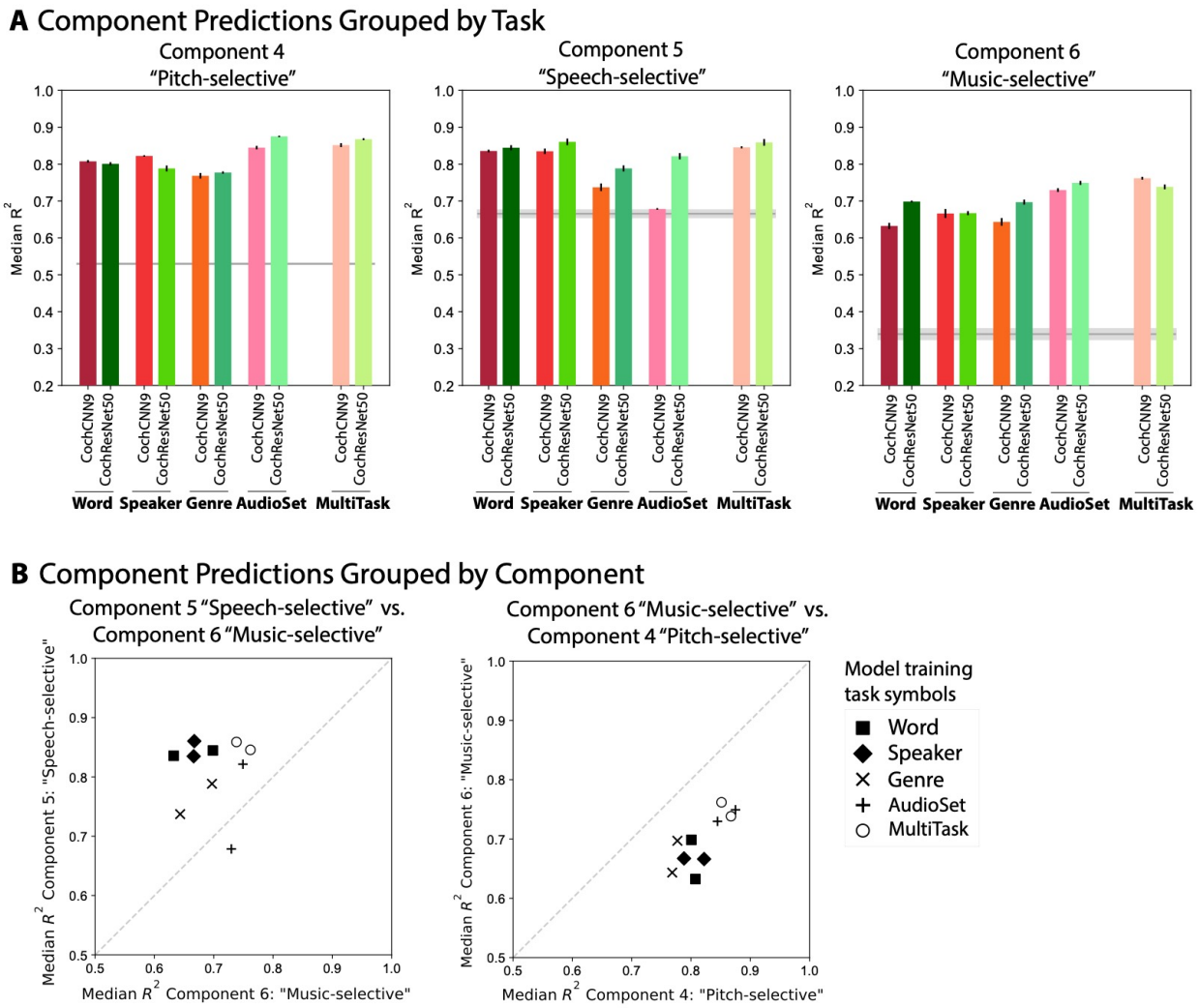


Figure 8. Training task modulates model predictions. (A) Component response variance explained by each of the trained in-house models. Predictions are shown for components 4-6 (pitch-selective, speech-selective, and music-selective, respectively). The in-house models were trained separately on each of four tasks as well as on three of the tasks simultaneously, using two different architectures. Explained variance was measured for the best-predicting stage of each model for each component selected using independent data. Error bars plot SEM over iterations of the model stage selection procedure (see Methods; Component modeling). Gray line plots the variance explained by the SpectroTemporal baseline model. **(B)** Scatter plots of in-house model predictions for pairs of components. The upper panel shows the variance explained for component 5 (speech-selective) vs. component 6 (music-selective), and the lower panels shows component 6 (music-selective) vs. component 4 (pitch-selective). Symbols denote the training task. In the left panel, the four models trained on speech-related tasks are furthest from the diagonal, indicating good predictions of speech-selective tuning at the expense of those for music-selective tuning. In the right panel, the models trained on the environmental sound task (AudioSet) are set apart from the others in their predictions of both the pitch-selective and music-selective components. Error bars are smaller than the symbol width (and are provided in panel A) and so are omitted for clarity.

We found that in each component and architecture, the multi-task models predicted component responses about as well as the best single-task model. It was not obvious a priori that a model trained on multiple tasks would capture the benefits of each single-task model – one might alternatively suppose that the demands of supporting multiple tasks with a single representation would muddy the ability to predict domain-specific brain responses. Indeed, the multi-task models achieved slightly lower task performance than the single-task models on each of the tasks (see Methods; Training CochResNet50 and CochCNN9 models – Word-Speaker-Noise tasks). This result is consistent with the results of Kell et al. that dual-task performance was impaired in models that were forced to share representations across tasks. However, the effect here was modest, and evidently did not prevent the multi-task model representations from capturing speech- and music-specific response properties. This result indicates that multi-task training is a promising path toward better models of the brain, in that the resulting models appear to combine the advantages of individual tasks.

Representation dimensionality correlates with model predictivity but does not explain it

Although the task manipulation showed a benefit of multiple tasks in our in-house models, the task alone does not obviously explain the large variance across external models in the measures of model-brain similarity that we used. Motivated by recent findings that the dimensionality of a model's representation tends to correlate with regression-based brain predictions of ventral visual cortex [48], we examined whether a model's effective dimensionality could account for some of the differences we observed between models (Supplementary Figure S6).

The effective dimensionality is intended to summarize the number of dimensions over which a model's activations vary for a stimulus set, and is estimated from the eigenvalues of the covariance matrix of the model activations to a stimulus set (see Methods; Effective dimensionality). Effective dimensionality is typically lower than a model's ambient dimensionality (i.e., the number of unit activations) because the activations of different units in a model can be correlated. Effective dimensionality must limit predictivity when a model's dimensionality is lower than the dimensionality of the underlying neural response, because a low dimensional model response could not account for all of the variance in a high dimensional brain response.

We measured effective dimensionality for each stage of each evaluated model (Supplementary Figure S6). We pre-processed the model activations to match the pre-processing used for the brain-model comparisons. The effective dimensionality for model stages ranged from ~1 to ~65 for our stimulus set (using the regression analysis pre-processing). By comparison, the effective dimensionality of the fMRI responses was 8.75 (for NH2015) and 5.32 (for B2021). Effective dimensionality tended to be higher in trained than in permuted models, and tended to increase from one model stage to the next in trained models. The effective dimensionality of a model stage was modestly correlated with the stage's explained variance (r^2=0.23 and 0.25 for NH2015 and B2021, respectively; Supplementary Figure S6, panel Aii), and with the model-brain RDM similarity (r^2=0.20 and 0.23 for NH2015 and B2021, respectively; Supplementary Figure S6, panel Bii). However, this correlation was much lower than the reliability of the explained variance measure (r^2=0.98, measured across the two fMRI data sets for trained networks; Supplementary Figure S6, panel Ai), and the reliability of the model-brain RDM similarity (r^2=0.96; Supplementary

Figure S6, panel Bi). Effective dimensionality thus does not explain the majority of the variance across models – there was wide variation in the dimensionality of models with good predictivity, and also wide variation in predictivity of models with similar dimensionality.

Intuitively, dimensionality could be viewed as a confound for regression-based brain predictions. High-dimensional model representations might be more likely to produce better regression scores by chance, on the grounds that the regression can pick out a small number of dimensions that approximate the function underlying the brain response, while ignoring other dimensions that are not brain-like. But because the RDM is a function of all of a representation's dimensions, it is not obvious why high-dimensionality on its own should lead to higher RDM similarity. Thus the comparable relationship between RDM similarity and dimensionality helps to rule out dimensionality as a confound in the regression analyses. In addition, both relationships were quite modest. Overall, the results show that there is a weak relationship between dimensionality and model-brain similarity, but that it cannot explain most of the variation we saw across models.

Discussion

We examined similarities between representations learned by contemporary deep neural network models and those in the human auditory cortex, using regression and representational similarity analyses to compare model and brain responses to natural sounds. We used two different brain data sets to evaluate a large set of models trained to perform audio tasks. Most of the models we evaluated produced more accurate brain predictions than a standard spectrotemporal filter model of the auditory cortex[32]. Predictions were consistently much worse for models with permuted weights, indicating a dependence on task-optimized features. The improvement in predictions with model optimization was particularly pronounced for cortical responses in non-primary auditory cortex selective for pitch, speech, or music. We observed task-specific prediction improvements for particular brain responses, e.g. with speech tasks producing the best predictions of speech-selective brain responses. Accordingly, the best overall predictions (aggregating across all voxels) were obtained with models trained on multiple tasks. We also found that most models exhibited correspondence with the presumptive auditory cortical hierarchy, with primary auditory voxels being best predicted by model stages that were consistently earlier than the best-predicting model stages for non-primary voxels. The training-dependent model-brain similarity and model-stage-brain-region correspondence was evident both with regression and representational similarity analyses. The results indicate that more often than not, deep neural network models optimized for audio tasks learn representations that capture aspects of human auditory cortical responses and organization.

Our general strategy was to test as many models as we could, and the model set included every audio model with an implementation in PyTorch that was publicly available at the time of our experiments. The benefit of this "kitchen sink" approach is that it provided a strong test of the generality of brain-DNN correspondence. The cost is that the resulting model comparisons were uncontrolled – the external models varied in architecture, training task, and training data, such that there is no way to attribute differences between model results to any one of these variables. To better distinguish the role of the training task, we complemented the external models with a set of models built in our lab that enabled a controlled manipulation of task. These models had

identical architectures, and for three of the tasks had the same training data, being distinguished only by which of three types of labels the model was asked to predict.

What do our results reveal about how to build a good model of human auditory cortex? First, they provide broad additional support for the idea that training a hierarchical model to perform tasks on natural, complex audio signals produces representations that exhibit some alignment with the cortex, better than was obtainable by previous generations of models. The fact that many models produce relatively good predictions suggests that these models contain audio features that typically align to some extent with brain representations, at least for the fMRI measurements we are working with. Second, some models built for engineering purposes produce poor brain predictions. Although the heterogeneity of the models limits our ability to diagnose the factors that underlie these brain-model discrepancies, the result is important insofar as it means that we should not expect every DNN model to produce strong alignment with the brain. Third, multiple tasks seem to improve predictions. Because the training stimuli for the two in-house architectures were the same for the multi-task model and for each of the corresponding three single-task models, the improvement from training with multiple tasks is clearly due to task rather than the data set. The results suggest that the benefit from training on multiple tasks is due to particular tasks producing representations that align well with particular brain responses, such that a model trained on multiple tasks gets the best of all worlds (Figure 8). Fourth, models with higher-dimensional representations are somewhat more likely to produce good matches with the brain. At present it is not clear what drives this effect, but there was a modest but consistent effect evident with both metrics we used. Fifth, the predictions from the best external models suggest that good predictions of fMRI responses can be obtained without providing a model with simulated cochlear input. Models built for engineering purposes typically operate on a spectrogram or raw waveform, and thus have initial processing stages that deviate from that of the human ear (whose tuning is broader for high than for low frequencies, for instance, unlike that of a conventional spectrogram). We have found in other work that building in a biologically faithful cochlear stage is important for replicating aspects of human auditory behavior (specifically that of pitch perception)[16], and so it seems likely that such a model stage will also prove essential for reproducing cortical responses in detail. However, the coarseness of fMRI responses may make the benefit of early-stage biological accuracy less evident.

Relation to prior work

The best-known prior study along these lines is that of Kell et al., (2018), and the results here qualitatively replicate the key results of that study. One contribution of the present study thus lies in showing that these earlier results hold for many different auditory models. In particular, most trained models produce better predictions than the SpectroTemporal baseline model, and most exhibit a correspondence between model stages and brain regions. The consistently worse predictions obtained from models with random/permuted weights also replicates prior work, providing more evidence that optimizing representations for tasks tends to bring them in closer alignment with biological sensory systems. In addition, we substantiated these main conclusions using representational similarity analyses in addition to regression-based predictions, providing converging evidence for model-brain matches. Overall, the results indicate a qualitatively similar

set of results to those obtained in the ventral visual pathway, where many different trained models produce overall good predictions[45].

The Kell et al. study used a model trained on two tasks, but did not test the extent to which the multiple tasks improved the overall match to human brain responses. Here we compared brain-model similarity for models trained on single tasks and models trained on multiple tasks, and saw advantages for multiple tasks. We note that it is not always straightforward to train models to perform multiple tasks, and indeed that the Kell et al. study found that task performance was optimized when the representations subserving the two tasks were partially segregated. This representational segregation could potentially interact with the extent to which the model representations match to human brain responses. But for the tasks we considered here, it was not necessary to explicitly force representational segregation in order to achieve good task performance, or good predictions of human brain responses.

Beyond the study by Kell et al., there have been relatively few other efforts to compare neural network models to auditory cortical responses. One study compared representational similarity of fMRI responses to music to the responses of a neural network trained on music annotations, but did not compare to standard baseline models of auditory cortex[20]. Another study optimized a network for a small-scale (10 digit) word recognition task, and reported seeing some neurophysiological properties of the auditory system[22]. Koumura et al.[21] trained networks on environmental sound or speech classification, and observed tuning to amplitude modulation, similar to that found in peripheral and mid-level stages of biological auditory systems, but did not investigate the putative hierarchy of cortical regions. Millet et al.[25] used a self-supervised speech model to predict brain responses to naturalistic speech, and found a stage-region correspondence similar to that in Kell et al. and the present work. However, the overall variance explained was very low. Similarly, Vaidya et al.[27] demonstrated that certain self-supervised speech models capture distinct stages of speech processing. Our results complement these findings in showing that they apply to a large set of models and to responses to natural sounds more generally.

The results here inform ongoing discussions regarding hierarchical computation in auditory cortex. The extent to which the cortex is organized into a sequence of stages corresponding to cortical regions is a central unresolved issue in human cognitive neuroscience[49–51,15]. Much of the evidence for hierarchy is associated with speech processing, in that speech-specific responses only emerge outside of primary cortical areas[52–57,33,58,41]. Other evidence for hierarchical organization comes from analyses of responses to natural sounds, which show selective responses to music and song in non-primary auditory cortex[33,34,59]. These non-primary responses occur with longer latencies and longer integration windows[60] than primary cortical responses. In addition, stimuli that are matched in audio and modulation frequency content to natural sounds drive responses in primary, but not non-primary, auditory cortex[61]. Non-primary areas also show invariance to real-world background noise[62]. The present results are consistent with all of these prior results in two main respects. First, they provide additional evidence for domain-specific responses (in that particular tasks produce model representations that best predict particular response components). Second, they provide further evidence for a broad distinction between the computational description of primary and non-primary auditory cortex (with primary and non-

primary voxels being consistently best-predicted by earlier and later stages of hierarchical models, respectively). We note that these results do not speak to the anatomical connections between regions, only to their stimulus selectivity and correspondence to hierarchical computational models. The present results in particular do not necessarily imply that the observed regional differences reflect strictly sequential stages of processing[63]. But they do show that the relationship to hierarchical model stages is fairly consistent across data sets and models.

Limitations

The analyses presented here are intrinsically limited by the coarseness of fMRI data in space and time. Voxels contain many thousands of neurons, and the slow time constant of the BOLD signal averages the underlying neuronal responses on a timescale of several seconds. It remains possible that responses of single neurons would be harder to relate to the responses of the sorts of models tested here, particularly when temporal dynamics are examined. Our analyses are also limited by the number of stimuli that can feasibly be presented in an fMRI experiment (less than 200 given our current methods and reliability standards). It is possible that larger stimulus sets would better differentiate the models we tested.

The conclusions here are also limited by the two metrics of model-brain similarity that we used. The regression-based metric of explained variance is based on the assumption that representational similarity can be meaningfully assessed using a linear mapping between responses to natural stimuli[64,38,65]. This assumption is common in systems neuroscience, but could obscure aspects of a model representation that deviate markedly from those of the brain, because the linear mapping picks out only the model features that are predictive of brain responses. There is ample evidence that deep neural network models tend to rely partially on different features than humans[66,67], as well as having partially distinct invariances[36,68] for reasons that remain unclear, and encoding model analyses likely mask the presence of such discrepant model properties. We note that accurate predictions of brain responses may be useful in a variety of applied contexts, and so have value independent of the extent to which they capture intuitive notions of similarity. In addition, accurate predictive models might be scientifically useful in helping to better understand what is represented in a brain response (e.g. by generating predictions of stimuli that yield high or low responses, that can then be tested experimentally[69]). But there are nonetheless limitations when relying exclusively on regression to evaluate whether a model replicates brain representations.

Representational dissimilarity matrices complement regression-based metrics, but have their own limitations. RDMs are computed from the entirety of a representation, and so reflect all of its dimensions, but conversely are not invariant to linear transformations. Scaling some dimensions up and others down can alter an RDM even if it does not alter the information that can be extracted from the underlying representation. Moreover, RDMs must be computed from sets of voxel responses, and so are sensitive to the (potentially ad hoc) choice of which voxels to pool together. For instance, our first analysis (Figure 2) pooled voxels across all of auditory cortex, and this may have limited the similarity observed with individual model stages. By contrast, regression metrics can be evaluated on individual voxels.

The fact that the regression and RDM analyses yielded similar qualitative conclusions is reassuring, but they are but two of a large space of possible metrics. In addition, the correspondence between the two metrics was not perfect. The correlation between overall variance explained (the regression metric) and the human-model RDM similarity across network stages was r^2=0.58 and 0.60 for NH2015 and B2021, respectively – much higher than chance, but below the noise ceiling for the two measures (Supplementary Figure S7). In addition, the best model stages for each ROI were generally weakly correlated between the two metrics (Figure 7). These discrepancies are not well understood at present, but must eventually be resolved for the modeling enterprise to declare success.

As discussed above, our study is unable to disentangle effects of architecture, task, and training data on the brain predictions of the external models tested. We emphasize that this was not our goal – we sought to test a wide range of models as a strong test of the generality of brain-DNN similarities, cognizant that this would limit our ability to assess the reasons for brain-model discrepancies. The in-house models nonetheless help reveal some of the factors that drive differences in model predictions.

Future directions

The finding that task-optimized neural networks generally enable improved predictions of auditory cortical brain responses motivates a broader examination of such models, as well as further model development for this purpose. For instance, the findings that different tasks best predict different brain responses suggest that models that both recognize and produce speech might help to explain differences in "dorsal" and "ventral" speech pathways[70], particularly if paired with branching architectures[15] that can be viewed as hypotheses for distinct functional pathways. Models trained to localize sounds[17] in addition to recognizing them might help explain aspects of the cortical encoding of sound location and its possible segregation from representations of sound identity[71–76]. Task-optimized models could potentially also help clarify findings that currently do not have an obvious functional interpretation, for instance the tendency for responses to broadband onsets to be anatomically segregated from responses to sustained and tonal sounds[33,77,78], if such response properties emerge for some tasks and not others.

The fact that optimizing models to perform tasks produces better brain predictions suggests that brain representations are shaped to some extent by task constraints. As is widely noted, the learning algorithm used in most of the models we considered (supervised learning) is not a plausible account for how task constraints might influence biological organisms[79]. The use of supervised learning is motivated by the possibility that one could converge on an accurate model of the brain's representations by replicating some constraints that shape neural representations even if the way those constraints are imposed deviates from biology. It is nonetheless conceivable (and perhaps likely) that fully accurate models will require learning algorithms that more closely resemble the optimization processes of biology, in which nested loops of evolutionary selection and (largely unsupervised) learning over development combine to produce systems that can perform a wide range of tasks with breathtaking accuracy and efficiency. Some steps in this direction can be found in recent models that are optimized without labeled training data[25,27,80,81]. Our model set contained one such contrastive self-supervised model (Wav2vec2), and although

its brain predictions were worse than those of most of the supervised models, this direction clearly merits extensive exploration.

It will also be important to use additional means of model evaluation, such as model-matched stimuli[61,36,68], stimuli optimized for the model's predicted response[82,83,69], or directly substituting brain responses into DNNs[84]. And ultimately, analyses such as these need to be related to more fine-grained anatomy and brain response measurements. Model-based analyses of human intracranial data[26,85] and single neuron responses from nonhuman animals both seem like promising next steps in the pursuit of complete models of biological auditory systems.

Methods

Methods

- Voxel response modeling
  - General
  - Voxelwise modeling: Regularized linear regression and cross-validation
  - Voxelwise modeling: Correcting for reliability of the measured voxel response
  - Voxelwise modeling: Correcting for reliability of the predicted voxel response
  - Voxelwise modeling: Corrected measure of variance explained
  - Voxelwise modeling: Summary
  - Voxelwise predictions across models [Figure 2]
  - Voxelwise predictions across model stages [Supplementary Figure S2]
  - Best-predicting model stage [Figure 6 / Figure 7]
- Component modeling
  - Component predictions across models [Figure 5 & Figure 8]
  - Component predictions across sounds [Figure 4]
- Representational Similarity Analysis
- Effective dimensionality
- Statistical analysis
  - Voxel responses: Pairwise model comparisons
  - Comparisons of best predicting model stages between ROIs
  - Component responses: Pairwise model comparisons
- fMRI data (NH2015)
  - fMRI cortical responses to natural sounds
  - Natural sound stimuli
  - fMRI scanning procedure
  - fMRI data acquisition
  - fMRI data preprocessing
  - Voxel selection
- fMRI data (B2021)
  - fMRI cortical responses to natural sounds
  - Natural sound stimuli
  - fMRI scanning procedure

- fMRI data acquisition
- fMRI data preprocessing
- Voxel selection
- Candidate models
  - External models
    - AST
    - DCASE2020 baseline
    - DeepSpeech2
    - MetricGAN
    - S2T
    - SepFormer
    - VGGish
    - VQ-VAE (ZeroSpeech2020)
    - Wav2vec 2.0
  - In-house models
    - Cochleagram inputs
    - SpectroTemporal model
    - CochCNN9 architecture
    - CochResNet50 architecture
    - Training data set for CochResNet50 and CochCNN9 models - Word-Speaker-Noise tasks
    - Training CochResNet50 and CochCNN9 models - Word-Speaker-Noise tasks
    - Training data set for CochResNet50 and CochCNN9 models - musical genre task
    - Training CochResNet50 and CochCNN9 models - musical genre task
  - Candidate models with permuted weights

Voxel response modeling

The following voxel encoding model methods are adapted from those of Kell et al., (2018) and where the methods are identical, we have reproduced the analogous sections of the methods verbatim. We summarize the minor differences from the Kell et al. methods at the end of this section. All voxel response modeling and analysis code was written in Python (version 3.6.10), making heavy use of the numpy[86] (version 1.19.0), scipy[87] (version 1.4.1), and scikit-learn[88] libraries (version 0.24.1).

General

We performed an encoding analysis in which each voxel's time-averaged activity was predicted by a regularized linear model of the DNN activity. We operationalized each model stage within each candidate model (see Section "Candidate models") as a hypothesis of a neural implementation of auditory processing. The fMRI hemodynamic signal to which we were comparing the candidate model blurs the temporal variation of the cortical response, thus a fair comparison of the model to the fMRI data involved predicting each voxel's time-averaged response to each sound from time-averaged model responses. We therefore averaged the model responses over the temporal dimension after extraction. Because it seemed possible that units with real-valued activations might average out to near-zero values, we extracted unit activations after model stages that transform the output to positive values (ReLU, Tanh stages). Transformer architectures had no such stages, so we extracted the real-valued unit activations, and analyzed all model stages in this way. Pilot analyses suggested that voxel predictions from these models were similar when we time-averaged unit activations that were exclusively positive (specifically, when we used the root-mean-square instead of the mean).

Voxelwise modeling: Regularized linear regression and cross-validation

We modeled each voxel's time-averaged response as a linear combination of a model stage's time-averaged unit responses. We first generated 10 randomly selected train/test splits of the 165 sound stimuli into 83 training sounds and 82 testing sounds. For each split, we estimated a linear map from model units to voxels on the 83 training stimuli and evaluated the quality of the prediction using the remaining 82 testing sounds (described below in greater detail). For each voxel-stage pair, we took the median across the 10 splits. The linear map was estimated using regularized linear regression. Given that the number of regressors (i.e., time-averaged model units) typically exceeded the number of sounds used for estimation (83), regularization was critical. We used L2-regularized (''ridge'') regression, which can be seen as placing a zero-mean Gaussian prior on the regression coefficients. Introducing the L2-penalty on the weights results in a closed-form solution to the regression problem, which is similar to the ordinary least-squares regression normal equation:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + n\lambda\mathbf{I})^{-\mathbf{1}}\mathbf{X}^T\mathbf{y}$$

where **w** is a d-length column vector (the number of regressors – i.e., the number of time-averaged units for the given stage), **y** is an n-length column vector containing the voxel's mean response to each sound (length 83), **X** is a matrix of regressors (n stimuli by d regressors), n is the number of stimuli used for estimation (83), and **I** is the identity matrix (d by d). We demeaned each column of the regressor matrix (i.e., each model unit's response to each sound), but we did not normalize the columns to have unit norm. Similarly, we demeaned the target vector (i.e., the voxel's or the component's response to each sound). By not constraining the norm of each column to be one, we implemented ridge regression with a non-isotropic prior on each unit's learned coefficient. Under such a prior, units with larger norm were expected a priori to contribute more to the voxel predictions. In pilot experiments, we found that this procedure led to more accurate and stable predictions in left-out data, compared with a procedure where the columns of the regressor

matrices were normalized (i.e., with an isotropic prior). The demeaning was performed on the train set and the same transformation was applied on the test set. This ensured independence (no data leakage) between the train and test sets.

Performing ridge regression requires selecting a regularization parameter that trades off between the fit to the (training) data and the penalty for weights with high coefficients. To select this regularization parameter, we used leave-one-out cross validation within the set of 83 training sounds. Specifically, for each of 100 logarithmically-spaced regularization parameter values (1e-50, 1e-49, …, 1e49, 1e50), we measured the squared error in the resulting prediction of the left out sound using regression weights derived from the other sounds in the training split. We computed the average of this error (across the 83 training sounds) for each of the 100 potential regularization parameter values. We then selected the regularization parameter that minimized this mean squared error. Finally, with the regularization parameter selected, we used all 83 training sounds to estimate a single linear mapping from a stage's features to a given voxel's response. We then used this linear mapping to predict the response to the left-out 82 test sounds, and evaluated the Pearson correlation of the predicted voxel response with the observed voxel response. If the predicted voxel response had a standard deviation of exactly zero (no variance of the prediction across test sounds), the Pearson correlation coefficient was set to 0. Similarly, if the Pearson correlation coefficient was negative, indicating that the held-out test sounds were not meaningfully predicted by the linear map from the training set, the Pearson correlation value was similarly set to 0. We squared this Pearson correlation coefficient to yield a measure of variance explained. We found that the selected regularization parameter values rarely fell on the boundaries of the search grid, suggesting that the range of the search grid was appropriate. We emphasize that the 82 test sounds on which predictions were ultimately evaluated were not incorporated into the procedure for selecting the regularization parameter nor for estimating the linear mapping from stage features to a voxel's response – i.e., the procedure was fully cross-validated.

Selecting regularization coefficients independently for each voxel-stage regression was computationally expensive, but seemed important for our scientific goals given that the optimal regularization parameter could vary across voxel-stage pairs. For instance, differences in the extent to which the singular value spectrum of the feature matrix is uniform or peaked (which influences the extent to which the $\mathbf{X}^T\mathbf{X} + n\lambda\mathbf{I}$ matrix in the normal equation above is well-conditioned) can lead to differences in the optimal amount of regularization. Measurement noise, which varies across voxels can also influence the degree of optimal regularization. By allowing different feature sets (stages) to have different regularization parameters we are enabling each feature set to make the best possible predictions, which is appealing given that the prediction quality is the critical dependent variable that we compare across voxels and stages. Varying the regularization parameter across feature sets while predicting the same voxel response will alter the statistics of the regression coefficients across feature sets, and thus would complicate the analysis and interpretation of regression coefficients. However, we are not analyzing the regression coefficients in this work.

Voxelwise modeling: Correcting for reliability of the measured voxel response

The use of explained variance as a metric for model evaluation is inevitably limited by measurement noise. To correct for the effects of measurement noise we computed the reliability of both the measured voxel response and the predicted voxel response. Correcting for the reliability of the measured response is important to make comparisons across different voxels, because (as shown in e.g., Figure S2 in Kell et al., (2018)) the reliability of the BOLD response varies across voxels. This variation can occur for a variety of reasons (e.g., distance from the head coil elements). Not correcting for the reliability of the measured response will downwardly bias the estimates of variance explained and will do so differentially across voxels. This differential downward bias could lead to incorrect inferences about how well a given set of model features explains the response of voxels in different parts of auditory cortex.

Voxelwise modeling: Correcting for reliability of the predicted voxel response

Measurement noise corrupts the test data to which model predictions are compared (which we accounted for by correcting for the reliability of the measured voxel response, as described above), but noise is also present in the training data and thus also inevitably corrupts the estimates of the regression weights mapping from model features to a given voxel. This second influence of measurement noise is often overlooked, but can be addressed by correcting for the reliability of the predicted response. Doing so is important for two reasons. First, as with the reliability of the measured voxel response, not correcting for the predicted voxel response can yield incorrect inferences about how well a model explains different voxels. Second, the reliability of the predicted response for a given voxel can vary across feature sets, and failing to account for these differences can lead to incorrect inferences about which set of features best explains that voxel's response. It was thus in practice important to correct for the reliability of the predicted voxel response. By correcting for both the reliability of the measured voxel response and the reliability of the predicted response, the ceiling of our measured r-squared values was 1 for all voxels and all stages, enabling comparisons of voxel predictions across all voxels and all neural network stages.

Voxelwise modeling: Corrected measure of variance explained

To correct for the reliability, we employ the correction for attenuation[43]. It is a standard technique in many fields, and is becoming more common in neuroscience. The correction estimates the correlation between two variables independent of measurement noise (here the measured voxel response and the model prediction of that response). The result is an unbiased estimator of the correlation coefficient that would be observed from noiseless data. Our corrected measure of variance explained was the following:

$$r^{2*}_{\mathbf{v},\hat{\mathbf{v}}} = \frac{\mathrm{r}(\mathbf{v}_{123}, \hat{\mathbf{v}}_{123})^2}{r'_{\mathbf{v}} r'_{\hat{\mathbf{v}}}}$$

where $\mathbf{v}_{123}$ is the voxel response to the 82 left-out sounds averaged over the three scans, $\hat{\mathbf{v}}_{123}$ is the predicted response to the 82 left-out sounds (with regression weights learned from the other 83 sounds), r is a function that computes the correlation coefficient, $r'_{\mathbf{v}}$ is the estimated reliability of that voxel's response to the 83 sounds and $r'_{\hat{\mathbf{v}}}$ is the estimated reliability of that predicted voxel's response. $r'_{\mathbf{v}}$ is the median of the correlation between all 3 pairs of scans (scan 0 with scan 1; scan 1 with scan 2; and scan 0 with scan 2), which is then Spearman-Brown corrected to account for the increased reliability that would be expected from tripling the amount of data [43]. $r'_{\hat{\mathbf{v}}}$ is analogously computed by taking the median of the correlations for all pairs of predicted responses (models fitted on a single scan) and Spearman-Brown correcting this measure. Note that for very noisy voxels, this division by the estimated reliability can be unstable and can cause for corrected variance explained measures that exceed one. To ameliorate this problem, we limited both the reliability of the prediction and the reliability of the voxel response to be greater than some value *k* [40]. For *k* = 1, the denominator would be constrained to always equal one and thus the ''corrected'' variance explained measured would be identical to uncorrected value. For *k* = 0, the corrected estimated variance explained measure is unaffected by the value *k*. This k-correction can be seen through the lens of a bias-variance tradeoff: this correction reduces the amount of variance in the estimate of variance explained across different splits of stimuli, but does it at the expense of a downward bias of those variance explained metrics (by inflating the reliability measure for unreliable voxels). For $r'_{\mathbf{v}}$, we used a k of 0.182, which is the $p < 0.05$ significance threshold for the correlation of two 83-dimensional Gaussian variables (i.e., with the same length as our 83-dimensional voxel response vectors used as the training set), while for $r'_{\hat{\mathbf{v}}}$ we used a k of 0.183 which is the $p < 0.05$ significance threshold for the correlation of two 82-dimensional Gaussian variables (i.e., same length as our 82-dimensional predicted voxel response vectors, the test set).

Voxelwise modeling: Summary

We repeated this procedure for each stage and voxel ten times, once each for 10 random train/test splits, and took the median explained variance across the ten splits for a given stage-voxel pair. We performed this procedure for all stages of all candidate models and all voxels (across two data sets: NH2015: 7694 voxels, B2021: 26,792 voxels). Thus, for each stage and voxel, this resulted in ten explained variance values (r2). We computed the median explained variance across these ten cross-validation splits for each voxel-stage pair. For comparison, we performed an identical procedure with the stages of a permuted network with the same architecture as our main networks (see Section "Candidate models with permuted weights") and the spectrotemporal baseline model. In all analyses, if a noise-corrected median explained variance value exceeded 1, we set the value to 1 to avoid an inflation of the explained variance.

In summary, the voxel prediction methods were largely the same as those in Kell et al., (2018), with the following differences. First, we imposed a different range of regularization constants to avoid hitting the bounds of the range. This difference was necessitated to accommodate a larger and more diverse set of models than in Kell et al. as well as changes to scikit learn in the years separating our study from that of Kell et al. Second, we set the r-squared values for negative r values to zero, rather than using signed r-squared values as in Kell et al. This seemed like the best choice given that negative r values indicates that a model cannot predict the data. Third, we

used a different limit for the reliability used to correct the explained variance. Our limit was the minimum correlation that would be statistically significant for a sample size of 82 and 83 (which is the sample size for which the reliability is measured), whereas Kell assumed a sample size of 165. Fourth, we omitted the Fisher z transform when averaging r-squared values, as it seemed hard to motivate. We strongly suspect that none of these differences qualitatively affect any important result, but list them here for transparency.

Voxelwise predictions across models [Figure 2]

To compare how well each candidate model explained the variance in the fMRI data, we aggregated the explained variance across all voxels in the data set of interest (NH2015: 7694 voxels, B2021: 26,792 voxels) for each model. We evaluated each candidate model using its best-predicting stage. Selection of the best-predicting model stage was performed in one of two ways. In the main analysis featured in Figure 2, for each voxel, we used half of the ten cross-validation test splits to select the best-predicting stage, and the remaining five test splits to obtain the median explained variance. This yielded a median explained variance per voxel. To ensure that this procedure did not depend on the random five cross-validation splits selected, we repeated this procedure ten times for each model. We then obtained the mean of the explained variance values for each voxel across these ten iterations. To mitigate concerns that this analysis might be affected by the overlap in sounds in the five splits used to select the best stage and the five splits used to measure the explained variance, we performed a second analysis in which we selected the best-predicting model stage using all the voxels for all but one participant, and then measured the explained variance in each of the voxels in the left-out participant. This analysis measures explained variance with data fully independent from that used to choose the best-stage, but is less consistent with the rest of the analyses (e.g., the maps of the best-predicting model stage, in which it was critical to choose the best-predicting stage separately for each voxel). We confirmed that the results shown in Figure 2 were qualitatively similar if this second procedure was used to choose the best-predicting stage for each model. To obtain an aggregated explained variance across voxels for each model, we first obtained the median across voxels within each participant, and then took the mean across participants. An identical procedure was used for the permuted networks.

Voxelwise predictions across model stages [Supplementary Figure S2]

To visualize how well each stage of each candidate model explained variance in the fMRI data, we aggregated the explained variance across all voxels in the data set of interest (NH2015: 7694 voxels, B2021: 26,792 voxels) for each model. Given that no model stage selection procedure took place, we simply obtained the median across voxels within each participant, and then took the mean across participants for each model stage, identical to the aggregation procedure for the best stage voxelwise predictions (Figure 2).

Best-predicting model stage [Figure 6 / Figure 7]

We also examined which model stage best predicted each voxel's response (an ''argmax'' analysis, i.e., the position that best predicts the response for each voxel). We assigned each

model stage a position index between 0 and 1 (using minmax normalization such that the first stage was assigned a value of 0 and the last stage a value of 1, i.e.: (current_stage - min_stage) / (num_stages - min_stage)). For summary maps (Figure 6), we predicted responses in individuals and then aggregated results across participants (median), after they were aligned in a common coordinate system (i.e., the FsAverage surface from FreeSurfer) using K-Nearest Neighbor interpolation. For the across-model summary map we took the median of the best model stage positions across the n=15 best-performing models (rounded to the first decimal place). The plots were visualized using Freeview using default parameters. The color overlay was an inverse color wheel. The color scale limits were set to extend from 0 to the stage beyond the most common best stage (across voxels in both fMRI data sets). The permuted control networks were visualized using an identical color scale to the trained networks.

To quantify these summary maps, we compared the best-predicting model stage within different regions of the auditory cortex (Figure 7). We used four anatomical region-of-interest (ROIs): one for primary auditory cortex along with three ROIs for posterior, lateral, and anterior non-primary auditory cortex. These ROIs were combinations of subsets of ROIs in the Glasser et al. parcellation [47]. We note that they were taken directly from a previous publication [34] where they were intended to capture the auditory cortical regions exhibiting reliable responses to natural sounds, and were not adapted in any way to the present analysis. For each model, we computed the relative model stage position of the best-predicting stage within each ROI (an ‘‘argmax’’ analysis, as shown on the summary maps, Figure 6), which we summarized by taking the median across voxels within each ROI for each participant followed by the mean across participants (similar to the aggregation procedure in Figure 2). This yielded an average relative best model stage position per candidate model within each ROI. An identical procedure was used for the permuted networks.

Component modeling

We complemented the voxelwise modeling with analogous predictions of components of the fMRI response derived from all of the voxels. In previous work [33] we found that voxel responses to natural sounds can be explained as a linear combination of six response components (Figure 3A). The components are derived from the auditory cortical voxels (pooled across participants) that exceed a criterion of reliability. Each component is defined by a response to each of the sounds in the stimulus set, and has a weight for each voxel in the pool.

We predicted the responses for each of these six components in a manner similar to the voxelwise modeling. The only difference was that we did not perform any noise-ceiling correction for the components (the components do not have repetitions across scan sessions, unlike the voxel responses). Thus, all component predictions reported are the “raw” explained variance (squared Pearson correlation coefficient).

Component predictions across models [Figure 5 & Figure 8]

To compare how well each candidate model explained the component responses, we selected the best-predicting model stage for each candidate model using independent data (identical to the procedure described in “Voxelwise predictions across models” for the voxel data).

Component predictions across sounds [Figure 4]

We visualized the component response predictions by plotting them against the actual responses (derived from Norman-Haignere et al., (2015)) for each sound. We did this for the best-predicting model stage of the CochResNet50-MultiTask (best-performing model overall, Figure 2A) for each component. The best-predicting model stage was selected across ten iterations of the independent model stage selection procedure, as described in Section “Voxelwise predictions across models”. For each of the ten iterations, we used the median explained variance value of five random cross-validation test splits to identify the best model stage. Thus, ten iterations of this procedure yielded ten best-predicting model stages. For each component, we selected the most frequently occurring best-predicting model stage as the stage with which to visualize a given component’s predictions. Given a component and model stage, we obtained the predicted component response for a sound by taking the mean over all cross-validation splits in which that sound was included in the test set. In that way, we obtained the average prediction for each sound and each component.

We note that the predicted component responses visualized in Figure 4 were demeaned during the regression procedure (using the training set mean to demean the test set, ensuring no data leakage between train/test sets, see Section: Voxelwise modeling: Regularized linear regression and cross validation) and are hence centered around 0 (ordinate values). The actual component responses in Figure 4 were taken directly from Norman-Haignere et al., (2015) without any transformations (abscissa values).

Representational Similarity Analysis

To assess the robustness of our conclusions to the evaluation metric, we also investigated the similarity of model and fMRI responses using Representational Similarity Analysis (RSA) [31,28,45]. We used the same set of model stages and time-averaged representations as were used in the regression-based voxelwise modeling analysis. To construct the model representational dissimilarity matrix (RDM) for each model and model stage, we computed the dissimilarity (1 minus the Pearson correlation coefficient) between the model activations evoked by each pair of sounds. Similarly, to construct the fMRI RDM, we computed the dissimilarity in voxel responses (1 minus the Pearson correlation coefficient) between all voxel responses from a participant to each pair of sounds. Before computing the RDMs from the fMRI or model responses, we z-scored the voxel or unit responses. As a measure of fMRI and model similarity, we computed the Spearman rank ordered correlation coefficient between the fMRI RDM and the model RDM.

Representational Similarity Analysis: Across model comparison

In the analysis shown in Figure 2, we compared the RDMs computed across all voxels of a participant to the RDM computed from the time-averaged unit responses of each stage of each model. To choose the best-matching stage, we first generated 10 randomly selected train/test splits of the 165 sound stimuli into 83 training sounds and 82 testing sounds. For each split, we computed the RDMs for each model stage and for each participant's fMRI data for the 83 training sounds. We then chose the model stage that yielded the highest Spearman ρ measured between the model stage RDM and the participant's fMRI RDM. Using this model stage, we measured the model and fMRI RDMs from the test sounds and computed the Spearman ρ. We repeated this procedure ten times, once each for 10 random train/test splits, and took the median Spearman ρ across the ten splits. We performed this procedure for all candidate models and all participants (across two data sets: NH2015: 8 participants, B2021: 20 participants) and computed the mean Spearman ρ across participants for each model. For comparison, we performed an identical procedure with permuted versions of each neural network model, and with the SpectroTemporal baseline model.

Representational Similarity Analysis: Noise ceiling

The representational similarity analysis is limited by measurement noise in the fMRI data. As an estimate of the RDM correlation that could be reasonably expected to be achieved between a model RDM and a single participant's fMRI RDM, we calculated the correlation between one participant's RDM and the average of all the other participant's RDM. Within each data set (NH2015 and B2021) we held out one participant and averaged the RDMs across the remaining participants. The RDMs were measured from the same 10 train/test splits of the 165 sounds described in the previous section, using the 82 test sounds for each split. We then calculated the Spearman ρ between the RDM from the held-out participant and the average participant RDM. We took the median Spearman ρ across the 10 splits of data to yield a single value for each participant. This procedure was repeated holding out each participant, and the noise ceiling shown in Figure 2B is the mean across the measured value for each held out participant. This corresponds to the "lower bound" of the noise ceiling used in prior work[28]. We plotted the noise ceiling on the results graphs rather than noise-correcting the human-model RDM correlation to be consistent with prior modeling papers that have used this analysis[28,45].

Representational Similarity Analysis: Best model stage analysis

We also examined which model stage best captured the RDM measured from each anatomical ROI (an ''argmax'' analysis). We assigned each model stage a position index between 0 and 1. Given that we only report the "argmax" for this analysis (and not the measured values), we used the full set of 165 sounds to compute the RDMs. For a given ROI, we measured each participant's fMRI RDM computed on the voxels within the ROI. For each model, we computed the RDM for each stage and measured the Spearman ρ between the model-stage RDM and the fMRI ROI RDM. We measured the argmax across the stages for each model and each participant. We then took the mean of this position index across participants to yield an average relative best model stage per candidate model within each ROI. An identical procedure was used for the permuted networks.

Representational Similarity Analysis: Visualization

For fMRI RDMs, we computed the RDM individually for each participant, and then averaged the RDM for visualization. Both fMRI and model RDMs are grouped and colored by the sound categories assigned in Norman-Haignere et al., (2015). fMRI RDMs for all auditory cortex voxels and for each ROI can be found in Supplementary Figure S1.

Effective dimensionality

We investigated the effective dimensionality (ED) of the representations at each model stage as well as of the measured fMRI activity. The ED[89] was calculated on the set of 165 sounds that were used for the brain comparisons. The ED was evaluated as:

$$\mathrm{ED} = \frac{(\Sigma_i \lambda_i)^2}{\Sigma_i \lambda_i^2}$$

where λ_i are the square of the singular values obtained from the matrix of <activations> by <sounds>. This matrix was measured from the fMRI or model activations that were used for the regression analysis (demeaning each voxel or unit response) or for the RSA analysis (z-scoring the voxel or unit responses). In practice, these two different forms of pre-processing altered the ED measure, with ED values being about twice as large following the RSA pre-processing compared to the regression pre-processing. For instance, the ED for the fMRI data was 8.75 (for NH2015) and 5.32 (for B2021) using the regression pre-processing, but 16.9 (NH2015) and 12.9 (B2021) when using the representational similarity pre-processing. We used the two types of pre-processing to maintain consistency with the two types of model-brain similarity analysis, but the conclusions of the ED analysis would not have been qualitatively different had we exclusively used one type of pre-processing or the other.

Statistical analysis

Voxel responses: Pairwise model comparisons

For statistical comparison of brain predictions between multi- and single-task models, we evaluated significance non-parametrically by bootstrapping across participants (n=8 for NH2015, n=20 for B2021). For each single-task model, we sampled the participant explained variance values with replacement (8 or 20 values, sampled 10,000 times) and took the average. The resulting histogram was compared to the multi-task model's observed value averaged across all participants. The p-value was obtained by counting the number of times the bootstrapped value was smaller than the observed value, divided by the number of bootstrap iterations (n=10,000). The test was one-tailed because it was motivated by the hypothesis that the multi-task model would produce better fMRI response predictions than the single-task models.

Comparisons of best predicting model stages between ROIs

For statistical comparison of the mean model stage position index for pairs of anatomical ROIs (related to Figure 7), we performed a Wilcoxon signed rank test. The test compared the average values across models obtained from the primary ROI versus the average values obtained from the non-primary ROI across models. The test was two-tailed.

Component responses: Pairwise model comparisons

For statistical comparison of component predictions between pairs of models (related to Figure 5 and 8), we evaluated significance non-parametrically via a permutation test. Based on the model stage selection procedure described in Section “Component predictions across models”, we obtained ten independently selected median explained variance values per component. For a given component, we took the average across the ten explained variance values for each model and then compute the difference between the two models. We generated a null distribution by randomly permuting the model assignment and measuring the difference between the average of these permuted model assignment lists. The p-value was obtained by counting the number of times the observed difference was smaller than the values measured from permuted data, divided by the number of permutations (n=10,000). The test was one-tailed because in each case it was motivated by a hypothesis that one model would produce better component response predictions than the other. Specifically, there were two types of comparisons. In the first, the trained models were compared to the SpectroTemporal baseline model. In the second, models trained on a task that was plausibly related to the selectivity of a component (e.g. the word task for the speech component) were compared to models trained on a task not obviously related to that component (e.g. the genre task for the speech component).

fMRI data (NH2015)

The initial sections of the fMRI data collection methods used in Norman-Haignere et al., (2015) are very similar to the methods reported in Kell et al., (2018) and the text is replicated with minor edits.

fMRI cortical responses to natural sounds

The fMRI data analyzed here is a subset of the data in Norman-Haignere et al., (2015), only including the participants who completed three scanning sessions. Eight participants (four female, mean age: 22 years, range: 19-25; all right-handed) completed three scanning sessions (each ~2 hours). Participants were non-musicians (no formal training in the five years preceding the scan), native English speakers, and had self-reported normal hearing. Two other participants only completed two scans and were excluded from these analyses, and three additional participants were excluded due to excessive head motion or inconsistent task performance. The decision to exclude these five participants was made before analyzing any of their fMRI data. All participants provided informed consent, and the Massachusetts Institute of Technology Committee on the Use of Humans as Experimental participants approved experiments.

Natural sound stimuli

The stimuli were a set of 165 two-second sounds selected to span the sorts of sounds that listeners most frequently encounter in day-to-day life (Norman-Haignere et al., 2015). All sounds were recognizable – i.e., classified correctly at least 80% of the time in a ten-way alternative

forced choice task run on Amazon Mechanical Turk, with 55-60 participants per sound. See Supplementary Table S1 for names of all stimuli and category assignments. To download all 165 sounds, see the McDermott lab website: http://mcdermottlab.mit.edu/ downloads.html.

fMRI scanning procedure

Sounds were presented using a block design. Each block included five presentations of the identical two-second sound clip. After each two-second sound, a single fMRI volume was collected (''sparse scanning''), such that sounds were not presented simultaneously with the scanner noise. Each acquisition lasted one second and stimuli were presented during a 2.4 s interval (200 ms of silence before and after each sound to minimize forward/backward masking by scanner noise). Each block lasted 17 s (five repetitions of a 3.4 s TR). This design was selected based on pilot results showing that it gave more reliable responses than an event-related design given the same amount of overall scan time. Blocks were grouped into eleven runs, each with fifteen stimulus blocks and four blocks of silence. Silence blocks were the same duration as the stimulus blocks and were spaced randomly throughout the run. Silence blocks were included to enable estimation of the baseline response. To encourage participants to attend equally to each sound, participants performed a sound intensity discrimination task. In each block, one of the five sounds was 7 dB lower than the other four (the quieter sound was never the first sound). Participants were instructed to press a button when they heard the quieter sound.

fMRI data acquisition

MR data were collected on a 3T Siemens Trio scanner with a 32-channel head coil at the Athinoula A. Martinos Imaging Center of the McGovern Institute for Brain Research at MIT. Each functional volume consisted of fifteen slices oriented parallel to the superior temporal plane, covering the portion of the temporal lobe superior to and including the superior temporal sulcus. Repetition time (TR) was 3.4 s (although acquisition time was only 1 s), echo time (TE) was 30 ms, and flip angle was 90 degrees. For each run, the five initial volumes were discarded to allow homogenization of the magnetic field. In-plane resolution was 2.1 x 2.1 mm (96 x 96 matrix), and slice thickness was 4 mm with a 10% gap, yielding a voxel size of 2.1 x 2.1 x 4.4 mm. iPAT was used to minimize acquisition time. T1-weighted anatomical images were collected in each participant (1mm isotropic voxels) for alignment and surface reconstruction.

fMRI data preprocessing

Functional volumes were preprocessed using FSL and in-house MATLAB scripts. Volumes were corrected for motion and slice time. Volumes were skull-stripped, and voxel time courses were linearly detrended. Each run was aligned to the anatomical volume using FLIRT and BBRegister[90,91]. These preprocessed functional volumes were then resampled to vertices on the reconstructed cortical surface computed via FreeSurfer[92], and were smoothed on the surface with a 3mm FWHM 2D Gaussian kernel to improve SNR. All analyses were done in this surface space, but for ease of discussion we refer to vertices as ''voxels'' throughout this paper. For each of the three scan sessions, we estimated the mean response of each voxel (in the surface space) to

each stimulus block by averaging the response of the second through the fifth acquisitions after the onset of each block (the first acquisition was excluded to account for the hemodynamic lag). Pilot analyses showed similar response estimates from a more traditional GLM. These signal-averaged responses were converted to percent signal change (PSC) by subtracting and dividing by each voxel's response to the blocks of silence. These PSC values were then downsampled from the surface space to a 2mm isotropic grid on the FreeSurfer-flattened cortical sheet. For summary maps, we registered each participant's surface to Freesurfer's fsaverage template.

Voxel selection

For individual participant analyses, we used the same voxel selection criterion as Kell et al., (2018), selecting voxels with a consistent response to sounds from a large anatomical constraint region encompassing the superior temporal and posterior parietal cortex. Specifically, we used two criteria: (1) a significant response to sounds compared with silence ($p < 0.001$, uncorrected); and (2) a reliable response to the pattern of 165 sounds across scans. The reliability measure was as follows:

$$r = 1 - \frac{\|\mathbf{v}_{12} - \mathrm{proj}_{\mathbf{v}3}\mathbf{v}_{12}\|_2}{\|\mathbf{v}_{12}\|_2}$$

$$\mathrm{proj}_{\mathbf{v}3}\mathbf{v}_{12} = \left(\frac{\mathbf{v}_3 \cdot \mathbf{v}_{12}}{\|\mathbf{v}_3\|_2^2}\right)\mathbf{v}_3$$

where $\mathbf{v}_{12}$ is the response of a single voxel to the 165 sounds averaged across the first two scans (a vector), and $\mathbf{v}_3$ is that same voxel's response measured in the third. The numerator in the second term in the first equation is the magnitude of the residual left in $\mathbf{v}_{12}$ after projecting out the response shared with $\mathbf{v}_3$. This ''residual magnitude'' is divided by its maximum possible value (the magnitude of $\mathbf{v}_{12}$). The measure is bounded between 0 and 1, but differs from a correlation in assigning high values to voxels with a consistent response to the sound set, even if the response does not vary substantially across sounds. We found that using a more traditional correlation-based reliability measure excluded many voxels in primary auditory cortex because some of them exhibit only modest response variation across natural sounds. We included voxels with a value of this modified reliability measure of 0.3 or higher, which when combined with the sound responsive t test yielded a total of 7694 voxels across the eight participants (mean number of voxels per participant: 961.75; range: 637-1221).

fMRI data (B2021)

fMRI cortical responses to natural sounds

The fMRI data analyzed here is from Boebinger et al., (2021)[34]. Twenty participants (fourteen female, mean age: 25 years, range: 18-34; all right-handed) completed three scanning sessions (each ~2 hours). Half of these participants (n=10) were highly-trained musicians, with an average of 16.3 years of formal training (ranging from 11-23 years, SD = 2.5) that began before the age of seven [93] and continued until the time of scanning. The other half of the participants (n=10) were

non-musicians with less than two years of total music training, which could not have occurred either before the age of seven or within the five years preceding the time of scanning. All participants provided informed consent, and the Massachusetts Institute of Technology Committee on the Use of Humans as Experimental Subjects approved experiments.

Natural sound stimuli

The stimuli consisted of the set of 165 two-second natural sounds from Norman-Haignere et al., (2015), as well as 27 additional music and drumming clips from a variety of musical cultures, for a total of 192 sounds. For consistency with the Norman-Haignere et al., (2015) data set, we constrained our analyses to the same set of 165 sounds.

fMRI scanning procedure

The fMRI scanning procedure was similar to the design of Norman-Haignere et al., (2015), except for the following minor differences. Each stimulus block consisted of three repetitions of an identical two-second sound clip, and lasted 10.2 s (three repetitions of a 3.4 s TR). Each of the three scanning sessions consisted of sixteen runs (for a total of 48 functional runs per participant), with each run containing twenty-four stimulus blocks and five silent blocks of equal duration that were evenly distributed throughout the run. The shorter stimulus blocks used in this experiment allowed each stimulus to be presented six times throughout the course of the 48 runs. To encourage participants to attend equally to each sound, participants performed a sound intensity discrimination task. In each block, either the second or third repetition was 12 dB lower, and participants were instructed to press a button when they heard the quieter sound.

fMRI data acquisition

The data acquisition parameters were similar to those from Norman-Haignere et al., (2015), with a few minor differences. MR data were collected on a 3T Siemens Prisma scanner with a 32-channel head coil at the Athinoula A. Martinos Imaging Center of the McGovern Institute for Brain Research at MIT. Each functional volume consisted of 48 slices oriented parallel to the superior temporal plane, covering the whole brain. However, all analyses were restricted to an anatomical mask encompassing the same portions of the temporal and parietal lobes as in Norman-Haignere et al., (2015). Repetition time (TR) was 3.4 s (TA = 1 s), echo time (TE) was 33 ms, and flip angle was 90 degrees. For each run, the four initial volumes were discarded to allow homogenization of the magnetic field. In-plane resolution was 2.1 x 2.1 mm (96 x 96 matrix), and slice thickness was 3 mm with a 10% gap, yielding a voxel size of 2.1 x 2.1 x 3.3 mm. An SMS acceleration factor of 4 was used in order to minimize acquisition time. T1-weighted anatomical images were collected in each participant (1mm isotropic voxels) for alignment and surface reconstruction.

fMRI data preprocessing

Preprocessing was identical to Norman-Haignere et al., (2015). However, the initial analyses of this data set differ from Norman-Haignere et al., (2015) in that a GLM was used to estimate voxel responses rather than signal averaging, which was necessary due to the use of shorter stimulus blocks that caused more overlap between BOLD responses to different stimuli. For each of the three scan sessions, we estimated the mean response of each voxel (in the surface space) by

modeling each block as a boxcar function convolved with a canonical hemodynamic response function (HRF). The model also included six motion regressors and a first-order polynomial noise regressor to account for linear drift in the baseline signal. The resulting voxel beta weights were then downsampled from the surface space to a 2mm isotropic grid on the FreeSurfer-flattened cortical sheet. For summary maps, we registered each participant's surface to Freesurfer's fsaverage template.

Voxel selection

The process of selecting voxels was identical to Norman-Haignere et al., (2015), except that the reliability of voxel responses was determined by comparing the vectors of 192 beta weights estimated separately for the two halves of the data (v1 = first three repetitions from runs 1-24, v2 = last three repetitions from runs 25-48). Voxels were selected using the following reliability measure:

$$r = 1 - \frac{\|\mathbf{v}_{12} - \mathrm{proj}_{\mathbf{v}3}\mathbf{v}_{12}\|_2^2}{\|\mathbf{v}_{12}\|_2}$$

$$\mathrm{proj}_{\mathbf{v}3}\mathbf{v}_{12} = \left(\frac{\mathbf{v}_3 \cdot \mathbf{v}_{12}}{\|\mathbf{v}_3\|_2^2}\right)\mathbf{v}_3$$

including voxels with a value of 0.3 or higher and further selecting only voxels with significant responses to sounds ($p < 0.001$, uncorrected). The combination of these two criteria yielded a total of 26,792 voxels across the twenty participants (mean number of voxels per participant: 1,340; range: 1,020 – 1,828).

Candidate models

We investigated a set of n=19 candidate models. Nine of these models were trained by other labs for engineering purposes ("external"), and ten of these models were trained by us ("in-house"). Table 1 and Table 2 below show an overview of the nine external models and ten in-house models, respectively. For completeness, the SpectroTemporal baseline model is included in Table 2 along the in-house models. Details on model architectures and training can be found below the tables.

Table 1. External model overview.

Model name	**Brief description**	**Model input**	**Model output**	**Training data set**
AST (Audio Spectrogram Transformer)[94]	Transformer architecture for audio classification.	Spectrogram	AudioSet label (527)	AudioSet (ImageNet pretraining)[37,95]

DCASE2020[96]	Recurrent network trained for automated audio captioning.	Spectrogram	Audio text captions (4,367)	Clotho V1[97]
DeepSpeech2[98]	Recurrent architecture for automatic speech recognition.	Spectrogram	Characters (29)	LibriSpeech[99]
MetricGAN[100]	Generative adversarial network for speech enhancement.	Spectrogram	Voice-enhanced audio	VoiceBank-DEMAND[101]
S2T (Speech-to-Text)[102]	Transformer architecture for automatic speech recognition and speech-to-text translation.	Spectrogram	Words (10,000)	LibriSpeech[99]
SepFormer (Separation Transformer)[103]	Transformer architecture for speech separation.	Waveform	Source-separated audio	WHAMR![104]
VGGish[105]	Convolutional architecture for audio classification.	Spectrogram	Video label (30,871)	YouTube-100M[105]
VQ-VAE (ZeroSpeech2020)[106]	Convolutional encoder architecture for speech synthesis in a target speaker's voice.	Spectrogram	Audio in target speaker's voice	ZeroSpeech 2019 training data set[107]
Wav2vec2[108]	Transformer architecture for automatic speech recognition.	Waveform	Characters (32)	LibriSpeech[99]

Table 2. In-house model overview.

Model name	**Brief description**	**Model input**	**Model output**	**Training data set**
CochCNN9 Word	Convolutional architecture for word recognition	Cochleagram	Word label (794)	Word-Speaker-Noise data set[36]
CochCNN9 Speaker	Convolutional architecture for	Cochleagram	Speaker label (433)	Word-Speaker-Noise

	speaker recognition			data set[36]
CochCNN9 AudioSet	Convolutional architecture for environmental sound classification (AudioSet)	Cochleagram	AudioSet label (517)	Word-Speaker-Noise data set[36]
CochCNN9 Multi-task	Convolutional architecture for word recognition, speaker recognition, and environmental sound classification (AudioSet)	Cochleagram	Three output layers: Word label (794), Speaker label (433), AudioSet label (517)	Word-Speaker-Noise data set[36]
CochCNN9 Genre	Convolutional architecture for music genre classification	Cochleagram	Genre label (41)	Genre task using MusicBrainz data[15]
CochResNet50 Word	Convolutional architecture for word recognition	Cochleagram	Word label (794)	Word-Speaker-Noise data set[36]
CochResNet50 Speaker	Convolutional architecture for speaker recognition	Cochleagram	Speaker label (433)	Word-Speaker-Noise data set[36]
CochResNet50 AudioSet	Convolutional architecture for environmental sound classification (AudioSet)	Cochleagram	AudioSet label (517)	Word-Speaker-Noise data set[36]
CochResNet50 Multi-task	Convolutional architecture for word recognition, speaker recognition, and environmental sound classification (AudioSet)	Cochleagram	Three output layers: Word label (794), Speaker label (433), AudioSet label (517)	Word-Speaker-Noise data set[36]
CochResNet50 Genre	Convolutional architecture for music genre classification	Cochleagram	Genre label (41)	Genre task using MusicBrainz data[15]
SpectroTemporal	Linear filterbank with spectral and temporal modulations	Cochleagram	Spectrotemporal embedding space	(None)

External models

Nine external models implemented in PyTorch were obtained from publicly available repositories. To accommodate the required dependencies, a separate software environment was created to run each model, and the versions of Python, PyTorch, and TorchAudio are reported separately for each model.

AST

Audio Spectrogram Transformer (AST) is an attention-based, convolution-free transformer architecture for audio classification.
We used the pretrained model available by Yuan Gong and colleagues as described in Gong et al., (2021)[94]. Specifically, we used the model that was pre-trained on ImageNet[95] using a vision transformer architecture (data-efficient image Transformer (DeiT)[109]) and afterwards trained on AudioSet[37] (the best single model checkpoint which consisted of a model where all weights were averaged across model checkpoints from the first to last training epoch, model name: "Full AudioSet, 10 tstride, 10 fstride, with Weight Averaging (0.459 mAP)" (https://github.com/YuanGongND/ast).

AST is composed of an initial embedding layer followed by 12 multi-level encoder blocks that match the transformer architecture[110,111]. Model activations were extracted at the output of each transformer encoder block. In addition to model activations from the transformer blocks, we extracted the initial embeddings that are fed to the model, as well as the final logits over AudioSet classes, yielding 14 layers in total.

As described in Gong et al., (2021), the audio input to AST is the raw audio waveform that is converted into a sequence of 128 log-mel filterbank features computed with 25 ms Hamming windows every 10 ms. As the model was trained on AudioSet, the input size to the model was 10.24s (1024 time frames). The model implementation zero-padded any input less than this length. Thus, the spectrogram was of size [1024, 128] ([n_temporal, n_spectral]), which in our analyses resulted from a zero-padded 2s audio clip. The spectrogram was normalized by subtracting the average value measured from the training data set spectrograms (in this case, AudioSet), and dividing by two times the training data set spectrogram standard deviation. The spectrogram was split into a sequence of 101 16x16 patches (see Gong et al., (2021) for details on the patch embedding procedure) with an overlap of 6 in both time and frequency (i.e., stride (10,10)). These patches were projected into an embedding of size 768 ("the patch embedding layer") using the single convolutional layers as specified under "Architecture" in Gong et al., (2021). Two classification tokens were prepended to the embedding, which was then passed through 12 transformer encoder blocks. AST was trained on the full AudioSet data set (consisting of the official balanced and full training set, i.e., around 2M segments) using cross-entropy. The final layer was a linear classification layer over 527 audio labels.

Architecture

The AST architecture is denoted below with the sizes of the tensors propagated through the network denoted in parentheses. Encoder refers to each transformer encoding block. Model stages that were used for voxel and component response modeling are denoted in bold.

Input	(1024,128)
Embedding: Conv2d(1, 768, kernel_size=(16, 16), stride=(10, 10))	**(12, 101, 768)**
Encoder_1	**(1214, 768)**
Encoder_2	**(1214, 768)**
Encoder_3	**(1214, 768)**
Encoder_4	**(1214, 768)**
Encoder_5	**(1214, 768)**
Encoder_6	**(1214, 768)**
Encoder_7	**(1214, 768)**
Encoder_8	**(1214, 768)**
Encoder_9	**(1214, 768)**
Encoder_10	**(1214, 768)**
Encoder_11	**(1214, 768)**
Encoder_12	**(1214, 768)**
Linear_1(in_features=768, out_features=527, bias=True)	**(1, 527)**

For AST, we thus extracted model representations from the following 14 layers with the number of unit activations (regressors) for each sound denoted in parentheses: Embedding (768), Encoder_1 (768), Encoder_2 (768), Encoder_3 (768), Encoder_4 (768), Encoder_5 (768), Encoder_6 (768), Encoder_7 (768), Encoder_8 (768), Encoder_9 (768), Encoder_10 (768), Encoder_11 (768), Encoder_12 (768), Linear_1 (527).

Extractions were performed using torch=1.8.1, torchaudio=0.8.1 in Python 3.8.11.

DCASE2020 baseline

The DCASE2020 baseline model (henceforth DCASE2020) is recurrent architecture trained for automated audio captioning[96], where the model accepts audio as input and outputs the textual description (i.e., the caption) of that signal. We used the pre-trained model implemented by Konstantinos Drossos and collaborators (https://github.com/audio-captioning/dcase-2020-baseline).

The input to the model is a log-mel spectrogram (audio was peak-normalized prior to spectrogram conversion, i.e., divided by the maximum value of the absolute value of the audio signal) with 64

frequency bins resulting from a short-time Fourier transform applying 23 ms windows (window size) every 11.5 ms (stride). This yields log spectrogram patches of 173 x 64 bins that are the inputs to the model, i.e., a 2D array [n_temporal, n_spectral]. These spectrograms are passed through a 3-layer bi-directional GRU encoder and a one bi-directional layer GRU decoder with a linear readout. There are residual connections between the second and third encoder GRUs. The linear readout is a linear projection into C classes representing the 4367 one-hot encoding of unique caption words. The decoder iterates for 22 time steps.

DCASE2020 was trained using cross-entropy loss on the development split of Clotho v1[97] which consists of 2893 audio clips with 14465 captions. The audio samples are of 15 to 30 seconds duration, each audio sample having five captions of length 8-20 words.

Architecture

The DCASE2020 architecture is denoted below with the sizes of the tensors propagated through the network denoted in parentheses. Model stages that were used for voxel and component response modeling are denoted in bold. The two outputs of bidirectional recurrent stages were concatenated (i.e., treated as different features).

Input	(173,64)
Dropout(p=0.25)	
GRU_1(input_size=64, output_size=256, bidirectional=True)	**(2,256)**
GRU_2(input_size=512, output_size=256, bidirectional=True)	**(2,256)**
GRU_3(input_size=512, output_size=256, bidirectional=True)	**(2,256)**
Dropout(p=0.25)	
GRU_4(input_size=512, output_size=256, bidirectional=False)	**(1,256)**
Linear_1(in_features=256, out_features=4367)	**(22, 4367)**

Thus, for DCASE, we extracted model representations from the following 5 layers with the number of unit activations (regressors) for each sound denoted in parentheses: GRU_1 (512), GRU_2 (512), GRU_3 (512), GRU_4 (256), Linear_1 (4367).

Extractions were performed using torch=1.3.1 in Python 3.7.10.

DeepSpeech2

DeepSpeech2 is a recurrent architecture for automatic speech recognition[98]. We used the pre-trained PyTorch model by Sean Naren and collaborators (https://github.com/SeanNaren/deepspeech.pytorch).

As described by Amodei et al., (2016)[98], the input to the model is a log-spectrogram with 161 frequency bins resulting from a short-time Fourier transform applying 20 ms windows (window

size) every 10 ms (stride). This yields log spectrogram patches of 201 x 161 bins that are the inputs to the model, i.e., a 2D array [n_temporal, n_spectral]. Each spectrogram was normalized by subtracting the mean spectrogram value and dividing by the standard deviation. These spectrograms were transformed by two 2D convolutional layers followed by five bidirectional recurrent Long Short-Term Memory (LSTM) layers and ending in a fully connected layer. The fully connected layer is a linear projection into C classes representing the vocabulary of the task. The vocabulary consists of 29 classes (output features), corresponding to English characters and space, apostrophe, blank. DeepSpeech2 was trained using a CTC loss on the Librispeech corpus[99] (960hrs).

Architecture

The DeepSpeech2 architecture is denoted below with the sizes of the tensors propagated through the network denoted in parentheses. Model stages that were used for voxel and component response modeling are denoted in bold. The two outputs of bidirectional recurrent stages (using the LSTM output cell states) were concatenated (i.e., treated as different features).

Input	(201, 161)
Conv2d_1(in_channels=1, out_channels=32, kernel_size=(41,11), stride=(2,2), padding=(20,5))	(32, 81, 101)
BatchNorm2d_1(num_features=32)	(32, 81, 101)
HardTanh_1(min_val=0, max_val=20)	**(32, 81, 101)**
Conv2d_2(in_channels=32, out_channels=32, kernel_size=(21,11), stride=(2,1), padding=(10,5))	(32, 41, 101)
BatchNorm2d_2(num_features=32)	(32, 41, 101)
HardTanh_2(min_val=0, max_val=20)	**(32, 41, 101)**
LSTM_1(input_size=1312, hidden_size=1024, bidirectional=True)	**(2, 1024)**
SequenceWise BatchNorm1d_1(num_features=1024)	(101, 1024)
LSTM_2(input_size=1024, hidden_size=1024, bidirectional=True)	**(2, 1024)**
SequenceWise BatchNorm1d_2(num_features=1024)	(101, 1024)
LSTM_3(input_size=1024, hidden_size=1024, bidirectional=True)	**(2, 1024)**
SequenceWise BatchNorm1d_3(num_features=1024)	(101, 1024)
LSTM_4(input_size=1024, hidden_size=1024, bidirectional=True)	**(2, 1024)**
SequenceWise BatchNorm1d_4(num_features=1024)	(101, 1024)
LSTM_5(input_size=1024, hidden_size=1024, bidirectional=True)	**(2, 1024)**
BatchNorm1d_5(num_features=1024)	(101, 1024)
Linear_1(in_features=1024, out_features=29	**(101, 29)**

Thus, for DeepSpeech2, we extracted model representations from the following 8 layers with the number of unit activations (regressors) for each sound denoted in parentheses: HardTanh_1 (2,592), HardTanh_2 (1,312), LSTM_1 (2,048), LSTM_2 (2,048), LSTM_3 (2,048), LSTM_4 (2,048), LSTM_5 (2,048), Linear_1 (29).

Extractions were performed using torch=1.7.1, torchaudio=0.7.2 in Python 3.6.13.

MetricGAN

MetricGAN+ (henceforth MetricGAN) is a generative adversarial network (GAN) for speech enhancement. We used the pretrained model available by SpeechBrain[112] (hosted by HuggingFace) as described in Fu et al., (2021)[100]. Specifically, we used a model that was pre-trained on the Voicebank-DEMAND data set[101] (training files: 20,000 (58.03hr) + validation files: 5,000 (14.65hr)) (https://huggingface.co/speechbrain/metricgan-plus-voicebank).

The generator of MetricGAN is a Bidrectional Long Short-Term Memory (BLSTM) with two bidirectional LSTM layers followed by two fully-connected layers. The objective of the generator is to estimate a mask consisting of the noise in the signal in order to generate clean speech. The discriminator of MetricGAN consists of a convolutional architecture (not investigated here).

As described in Fu et al., (2021), the audio input to the model is the magnitude spectrogram resulting from a short-time Fourier transform applying 32 ms (window size) windows every 16 ms (stride) resulting in 256 power frequency bins. This yields magnitude spectrogram patches of 126 x 256 bins that are the inputs to the model, i.e., a 2D array [n_temporal, n_spectral] that are passed through the BLSTM and linear layers of the generator model.

Architecture

The MetricGAN architecture is denoted below with the sizes of the tensors propagated through the network denoted in parentheses. Model stages that were used for voxel and component response modeling are denoted in bold. The two outputs of bidirectional recurrent stages (using the LSTM output cell states) were concatenated (i.e., treated as different features).

Input	**(126, 257)**
LSTM_1(input_size=257, hidden_size=200, bidirectional=True)	**(2, 200)**
LSTM_2(input_size=257, hidden_size=200, bidirectional=True)	**(2, 200)**
Linear_1(in_features=400, out_features=300, bias=True)	(126, 300)
LeakyReLU_1	**(126, 300)**
Linear_2(in_features=300, out_features=257, bias=True)	**(126, 257)**

For MetricGAN, we thus extracted model representations from the following 4 layers with the number of unit activations (regressors) for each sound denoted in parentheses: LSTM_1 (400), LSTM_2 (400), LeakyReLU_1 (300), Linear_2 (257).

Extractions were performed using torch=1.9.1, speechbrain=0.5.10, huggingface-hub=0.0.17 in Python 3.8.11.

S2T

S2T (also known as Speech-to-Text) is an attention-based transformer architecture for automatic speech recognition (ASR) and speech-to-text translation (ST). We used the pre-trained model available by HuggingFace[113] as described in Wang et al., (2020)[102]. Specifically, we used the large model trained on Librispeech corpus[99] (960hrs) (https://huggingface.co/facebook/s2t-large-librispeech-asr).

S2T is an encoder-decoder model. The encoder part is composed of two convolutional layers followed by 12 multi-level encoder blocks that match the transformer architecture[110,111]. Model activations were extracted at the output of each transformer encoder block. In addition to model activations from the transformer blocks, we extracted the initial embeddings that were fed to the model, yielding 13 layers in total. We did not investigate the decoder part of the model.

As described by Wang et al., (2020), the audio input to S2T is a log-mel spectrogram with 80 mel-spaced frequency bins resulting from a short-time Fourier transform applying 25 ms windows every 10 ms. Each spectrogram was normalized by subtracting the mean value of the spectrogram and dividing by the standard deviation. This yields the log-mel spectrogram of 198 x 80 bins that are the inputs to the model, i.e., a 2D array [n_temporal, n_spectral]. The spectrogram is passed through two convolutional layers before it is then passed through the 12 transformer encoder blocks. S2T was trained using cross-entropy loss, and the output consists of the 10K unigram vocabulary from SentencePiece[114].

Architecture

The S2T architecture is denoted below with the sizes of the tensors propagated through the network denoted in parentheses (which is determined by the total stride in the initial feature encoder part of the architecture; not investigated here). Encoder refers to each transformer encoding block. Model stages that were used for voxel and component response modeling are denoted in bold.

Input	(198, 80)
Embedding / input post feature encoder	**(50, 1024)**
Encoder_1	**(50, 1024)**
Encoder_2	**(50, 1024)**
Encoder_3	**(50, 1024)**

Encoder_4	**(50, 1024)**
Encoder_5	**(50, 1024)**
Encoder_6	**(50, 1024)**
Encoder_7	**(50, 1024)**
Encoder_8	**(50, 1024)**
Encoder_9	**(50, 1024)**
Encoder_10	**(50, 1024)**
Encoder_11	**(50, 1024)**
Encoder_12	**(50, 1024)**

Thus, for S2T, we extracted model representations from the following 13 layers with the number of unit activations (regressors) for each sound denoted in parentheses: Embedding (1024), Encoder_1 (1024), Encoder_2 (1024), Encoder_3 (1024), Encoder_4 (1024), Encoder_5 (1024), Encoder_6 (1024), Encoder_7 (1024), Encoder_8 (1024), Encoder_9 (1024), Encoder_10 (1024), Encoder_11 (1024), Encoder_12 (1024).

Extractions were performed using transformers=4.10.0, torch=1.9.0, huggingface-hub=0.0.16 in Python 3.8.11.

SepFormer

SepFormer (also known as Separation Transformer) is an attention-based transformer architecture for speech separation. We used the pretrained model available by SpeechBrain[112] (hosted by HuggingFace) as described in Subakan et al., (2021)[103]. Specifically, we used a model that was pre-trained on the WHAMR! data set[104] (training files: 20,000 (58.03hr) + validation files: 5,000 (14.65hr)) (https://huggingface.co/speechbrain/sepformer-whamr).

SepFormer is composed of an initial encoder followed by 32 multi-level dual-path encoder blocks similar to the transformer architecture[110,111]. followed by a decoder. The transformer blocks follow a dual-path framework consisting of transformer blocks that model the short-term dependencies (IntraTransformer, IntraT), and Transformer blocks that model longer-term dependencies (InterTransformer, InterT). There are respectively 8 such IntraT and InterT blocks, yielding 16 transformer blocks, which is then repeated twice, yielding 32 transformer blocks in total. The objective of the dual-path transformer architecture is to estimate optimal masks to separate the audio sources present in the audio mixtures. The model was trained using scale-invariant source-to-noise ratio (SI-SNR) loss.

As described in Subakan et al., (2021), the audio input to AST is the raw audio waveform that is transformed by a single convolutional layer (encoder) followed by chunking the temporal

dimension into patches of size 250. These chunks were then passed through the 32 transformer encoder blocks.
Model activations were extracted at the output of each transformer encoder block. In addition to model activations from the transformer blocks, we extracted the initial encoder embeddings that are fed to the model, yielding 33 layers in total.

Architecture
The SepFormer architecture is denoted below with the sizes of the tensors propagated through the network denoted in parentheses. Encoder refers to each transformer encoding block. Model stages that were used for voxel and component response modeling are denoted in bold.

Input	(1, 16000)
Embedding: Conv1d(1, 256, kernel_size=(16,), stride=(8,), bias=False)	(256, 1999)
ReLU_1()	**(256, 1999)**
Conv1d(256, 256, kernel_size=(1,), stride=(1,), bias=False)	(256, 1999)
Encoder_1	**(18, 250, 256)**
Encoder_2	**(18, 250, 256)**
…	**…**
Encoder_31	**(18, 250, 256)**
Encoder_32	**(18, 250, 256)**

For SepFormer, we thus extracted model representations from the following 33 layers with the number of unit activations (regressors) for each sound denoted in parentheses: Embedding (after ReLU) (256), Encoder_1 (256), Encoder_2 (256), …, Encoder_31 (256), Encoder_32 (256).

Extractions were performed using torch=1.9.1, speechbrain=0.5.10, huggingface-hub=0.0.17 in Python 3.8.11.

VGGish

VGGish is a convolutional architecture for audio classification inspired by the VGG model for image recognition[115]. VGGish converts audio input features into a semantically meaningful, 128-dimensional embedding. We used the pretrained VGGish by Hershey et al., (2017)[105] (https://github.com/tensorflow/models/tree/master/research/audioset, specifically the PyTorch-compatible port by Harri Taylor and collaborators as found here: https://github.com/harritaylor/torchvggish).

VGGish was trained on the YouTube-100M corpus (70M training videos, 5.24 million hours with 30,871 labels) [105]. The videos average 4.6 minutes and are (machine) labeled with 5 labels on

average per video from the set of 30,871 labels. The model was trained to predict the video-level labels based on audio information using a cross-entropy loss function. As described by Hershey et al., (2017), the audio input consists of 960 ms audio frames that are decomposed with a short-time Fourier transform applying 25 ms (window size) windows every 10 ms (stride) resulting in 64 log mel-spaced frequency bins. This yields log-mel spectrogram patches of 96 x 64 bins that are the inputs to the model, i.e., a 3D array [n_frames, n_temporal, n_spectral]. Given that VGGish contained an additional temporal dimension (the n_frames dimension), we averaged over both temporal dimensions (the n_temporal dimension which corresponds to the time dimension of the spectrogram as well as the n_frames dimension which corresponds to the batch dimension) to obtain a time-averaged model representation.

Architecture

The VGGish architecture is denoted below with the sizes of the tensors propagated through the network denoted in parentheses. Model stages that were used for voxel and component response modeling are denoted in bold.

Input	(2, 96, 64)
Conv2d_1(in_channels=1, out_channels=64, kernel_size=(3,3), stride=(1,1), padding=(1,1))	(2, 64, 96, 64)
ReLU_1()	**(2, 64, 96, 64)**
MaxPool2d_1(kernel_size=2, stride=2, padding=0)	**(2, 64, 48, 32)**
Conv2d_2(in_channels=64, out_channels=128, kernel_size=(3,3), stride=(1,1), padding=(1,1)	(2, 128, 48, 32)
ReLU_2()	**(2, 128, 48, 32)**
MaxPool2d_2(kernel_size=2, stride=2, padding=0)	**(2, 128, 24, 16)**
Conv2d_3(in_channels=128, out_channels=256, kernel_size=(3,3), stride=(1,1), padding=(1,1))	(2, 256, 24, 16)
ReLU_3()	**(2, 256, 24, 16)**
Conv2d_4(in_channels=256, out_channels=256, kernel_size=(3,3), stride=(1,1), padding=(1,1)	(2, 256, 24, 16)
ReLU_4()	**(2, 256, 24, 16)**
MaxPool2d_4(kernel_size=2, stride=2, padding=0)	**(2, 256, 12, 8)**
Conv2d_5(in_channels=256, out_channels=512, kernel_size=(3,3), stride=(1,1), padding=(1,1))	(2, 512, 12, 8)
ReLU_5()	**(2, 512, 12, 8)**
Conv2d_6(in_channels=512, out_channels=512, kernel_size=(3,3), stride=(1,1), padding=(1,1))	(2, 512, 12, 8)

ReLU_6()	**(2, 512, 12, 8)**
MaxPool2d_6(kernel_size=2, stride=2, padding=0)	**(2, 512, 6, 4)**
Linear_1(in_features=12288, out_features=4096)	(2, 4096)
ReLU_7()	**(2, 4096)**
Linear_2(in_features=4096, out_features=4096)	(2, 4096)
ReLU_8()	**(2, 4096)**
Linear_3(in_features=4096, out_features=128)	(2, 128)
ReLU_9()	**(2, 128)**

Thus, for VGGish, we extracted model representations from the following 13 layers with the number of unit activations (regressors) for each sound denoted in parentheses: ReLU_1 (4,096), MaxPool2d_1 (2,048), ReLU_2 (4,096), MaxPool2d_2 (2,048), ReLU_3 (4,096), ReLU_4 (4,096), MaxPool2d_3 (2,048), ReLU_5 (4,096), ReLU_6 (4,096), MaxPool2d_4 (2,048), ReLU_7 (4,096), ReLU_8 (4,096), ReLU_9 (128).

Extractions were performed using torch=1.8.0 and torchaudio=0.8.1 in Python 3.8.5.

VQ-VAE (ZeroSpeech2020)

Vector-quantized variational autoencoder (henceforth VQ-VAE) is an encoder-decoder architecture trained to synthesize speech in a target speaker's voice. The model was trained for the ZeroSpeech 2020 challenge[116]. We used the pretrained model by Benjamin van Niekerk and colleagues as described in Niekerk et al., (2020)[106] (https://github.com/bshall/ZeroSpeech).

VQ-VAE consists of a CNN-based encoder and an RNN-based decoder. The encoder encodes the audio spectrogram and the decoder produces the new sound waveform. The model maps speech into a discrete latent space before reconstructing the original waveform.

As described by Niekerk et al., (2020), the input to VQ-VAE is the log-mel spectrogram (audio was peak-normalized prior to spectrogram conversion by dividing by the maximum of the absolute value of the audio signal, and this signal was multiplied by 0.999) with 80 mel-spaced frequency bins resulting from a short-time Fourier transform applying 25 ms windows every 10 ms. This yields the log-mel spectrogram of 201 x 80 bins that are the inputs to the model, i.e., a 2D array [n_temporal, n_spectral]. These spectrograms were transformed by five 1D convolutional layers. The model was trained to maximize the log-likelihood of the waveform given the discretized latent space bottleneck (details in Niekerk et al., (2020)). The model was trained on the ZeroSpeech 2019 English data set consisting of the Train Voice Dataset (4h40min) and the Train Unit Dataset (15h40min)[107]. To extract model activations, the audio samples were converted to the first encountered speaker ID on the available speaker ID list ("S015").

Architecture

The VQ-VAE encoder architecture is denoted below with the sizes of the tensors propagated through the network denoted in parentheses. We did not investigate the decoder part of VQ-VAE. Model stages that were used for voxel and component response modeling are denoted in bold.

Input	(80,201)
Conv1d_1(80, 768, kernel_size=(3,), stride=(1,), bias=False)	(768,199)
BatchNorm1d_1(768, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	(768,199)
ReLU_1()	**(768,199)**
Conv1d(768, 768, kernel_size=(3,), stride=(1,), padding=(1,), bias=False)	(768,199)
BatchNorm1d(768, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	(768,199)
ReLU_2()	**(768,199)**
Conv1d(768, 768, kernel_size=(4,), stride=(2,), padding=(1,), bias=False)	(768,99)
BatchNorm1d(768, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	(768,99)
ReLU_3()	**(768,99)**
Conv1d(768, 768, kernel_size=(3,), stride=(1,), padding=(1,), bias=False)	(768,99)
BatchNorm1d(768, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	(768,99)
ReLU_4()	**(768,99)**
Conv1d(768, 768, kernel_size=(3,), stride=(1,), padding=(1,), bias=False)	(768,99)
BatchNorm1d(768, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	(768,99)
ReLU_5()	**(768,99)**

Thus, for VQ-VAE, we extracted model representations from the following 5 layers with the number of unit activations (regressors) for each sound denoted in parentheses: ReLU_1 (768), ReLU_2 (768), ReLU_3 (768), ReLU_4 (768), ReLU_5 (768).

Extractions were performed using torch=1.9.0 in Python 3.8.11.

Wav2vec 2.0

Wav2vec 2.0 (henceforth Wav2vec2) is a self-supervised transformer architecture for automatic speech recognition that learns representations of speech from masked parts of raw audio. We used the pretrained model from Huggingface Transformers[113]; original model can be found here: https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec#wav2vec-20). Specifically, we used the base version trained and fine-tuned on the Librispeech corpus[99] (960hrs) (https://huggingface.co/facebook/wav2vec2-base-960h).

Wav2vec2 is composed of an initial multi-layer convolutional feature encoder followed by 12 multi-level encoder blocks that match the transformer architecture[110,111]. Model activations were extracted at the output of each transformer encoder block. In addition to model activations from the transformer blocks, we extracted the initial embeddings that are fed to the model, as well as the final logits over character tokens, yielding 14 layers in total.

As described by Baevski et al., (2020)[108] the audio input to Wav2vec2 is a sound waveform of zero mean and unit variance. Wav2vec2 is trained via a contrastive task where the true speech input is masked in a latent space and has to be distinguished from distractors. The contrastive loss is augmented by a diversity loss to encourage the model to use samples equally often. The pre-trained model is fine-tuned for speech recognition by adding a linear projection on top of the network into C classes representing the vocabulary of the task by minimizing a CTC loss [117]. The vocabulary consists of 32 classes (output features), corresponding to English characters + bos_token='<s>', eos_token='</s>', unk_token='<unk>', pad_token='<pad>', word_delimiter_token='|'.

Architecture

The Wav2vec2 architecture is denoted below with the sizes of the tensors propagated through the network denoted in parentheses (which is determined by the total stride in the initial feature encoder part of the architecture; not investigated here). Encoder refers to each transformer encoding block. Model stages that were used for voxel and component response modeling are denoted in bold.

Input	(32000)
Embedding / input post feature encoder	**(99, 768)**
Encoder_1	**(99, 768)**
Encoder_2	**(99, 768)**
Encoder_3	**(99, 768)**
Encoder_4	**(99, 768)**
Encoder_5	**(99, 768)**
Encoder_6	**(99, 768)**
Encoder_7	**(99, 768)**

Encoder_8	**(99, 768)**
Encoder_9	**(99, 768)**
Encoder_10	**(99, 768)**
Encoder_11	**(99, 768)**
Encoder_12	**(99, 768)**
Linear_1	**(99, 768)**

For Wav2vec2, we thus extracted model representations from the following 14 layers with the number of unit activations (regressors) for each sound denoted in parentheses: Embedding (768), Encoder_1 (768), Encoder_2 (768), Encoder_3 (768), Encoder_4 (768), Encoder_5 (768), Encoder_6 (768), Encoder_7 (768), Encoder_8 (768), Encoder_9 (768), Encoder_10 (768), Encoder_11 (768), Encoder_12 (768), Linear_1 (32).

Extractions were performed using transformers=4.5.0, torch=1.8.1 in Python 3.8.8.

In-house models

The in-house models consisted of a fixed cochleagram stage followed by either a convolutional architecture similar to that used in Kell et al., (2018) or a ResNet50 architecture. We refer to the full model architectures as CochCNN9 (indicating the 9 stages of this model) and CochResNet50. The models were trained either on the Word-Speaker-Noise data set[36], which supports three different tasks (word, speaker and environmental sound recognition), or the musical genre data set compiled in Kell et al., (2018). In house models were trained and evaluated with Python 3.8.2 and PyTorch 1.5.0.

Cochleagram inputs

The SpectroTemporal model and all CochResNet50 and CochCNN9 architectures had a cochleagram representation as the input to the model. A cochleagram is a time-frequency representation of the audio with frequency bandwidth and spacing that mimics the human ear, followed by a compressive nonlinearity[118,119]. The audio waveform passes through a bank of 211 bandpass filters ranging from 50Hz to 10kHz. Audio was sampled at 20kHz for the SpectroTemporal and the Word-Speaker-Noise task models, and was sampled at 16kHz for the genre task models. Filters are zero-phase with frequency response equal to the positive portion of a single period of a cosine function. Filter spacing was set by the Equivalent Rectangular Bandwidth (ERB_N) scale. Filters perfectly tile the spectrum such that the summed square response across all frequencies is flat, which includes four low-pass and four high-pass filters. The envelope was extracted from each filter subband using the magnitude of the analytic signal (Hilbert transform), and the envelopes were raised to the power of 0.3 to simulate basilar membrane compression. The resulting envelopes were lowpass filtered and downsampled to

200Hz, without any zero padding, resulting in a cochleagram representation of 211 frequency channels by 390 time points. This representation was the input to the auditory models. Cochleagram generation was implemented in PyTorch (code available: https://github.com/jenellefeather/chcochleagram).

SpectroTemporal model

For comparison to previous hand-engineered models of the auditory system we included a single layer SpectroTemporal model based on Chi et al., (2005)[32]. The main difference was that spectral filters were specified in cycles/erb (rather than cycles/octave) as the input signal to the model is a cochleagram with ERB-spaced filters. The model consists of a linear filter bank tuned to spectrotemporal modulations at different frequencies, spectral scales, and temporal rates. The different frequencies were implemented via applying the spectrotemporal filters as a 2D convolution with zero padding in frequency (800 samples) and time (211 samples). Spectrotemporal filters were constructed with center frequencies for the spectral modulations of [0.0625, 0.125, 0.25, 0.5, 1, 2] cycles/erb. Center frequencies for the temporal modulations consisted of [0.5, 1, 2, 4, 8, 16, 32, 64] and both upward and downward frequency modulations were included (resulting in 96 filters). An additional 6 purely spectral and 8 purely temporal modulation filters were included for a total of 110 modulation filters. To extract the power in each frequency band for each filter, we squared the output of each filter response at each time step and took the average across time for each frequency channel, similar to previous studies [15,61]. These power measurements were used as the regressors for voxel and component modeling (23421 activations).

CochCNN9 architecture

The CochCNN9 architecture is based on the architecture in Kell et al., (2018) that emerged from a neural network architecture search. The architecture used here differed in that the input to the first layer of the network is maintained as the 211x390 size cochleagram rather than being reshaped to 256x256. The convolutional layer filters and pooling regions were adjusted from those of the Kell et al. architecture to maintain the same receptive field size in frequency and time given the altered input dimensions. The other difference was that the network here was trained with batch normalization rather than the local response normalization used in Kell et al., (2018). Along with the CochResNet50, this architecture was used for task-optimization comparisons throughout the paper.

The CochCNN9 architecture is denoted below with the sizes of the tensors propagated through the network denoted in parentheses. Model stages that were used for voxel and component response modeling are denoted in bold.

Input (cochleagram)	**(1,211,390)**
BatchNorm2d_1 (1)	(1,211,390)

Conv2d_1(1, 96, kernel_size = [7, 14], stride = [3, 3], padding = 'same')	(96, 71, 130)
ReLU_1	**(96, 71, 130)**
MaxPool2d_1(kernel_size = [2,5] , stride = [2,2], padding = 'same')	**(96, 36, 65)**
BatchNorm2d_2(96)	(96, 36, 65)
Conv2d_2(96, 256, kernel_size = [4,8], stride = [2,2], padding = 'same')	(256, 18, 33)
ReLU_2	**(256, 18, 33)**
MaxPool2d_2(kernel_size = [2,5] , stride = [2,2], padding = 'same')	**(256, 9, 17)**
BatchNorm2d_3 (256)	(256, 9, 17)
Conv2d_3 (512, kernel_size = [2,5], stride = [1,1], padding = 'same')	(512, 9, 17)
ReLU_3	**(512, 9, 17)**
Conv2d_4 (1024, kernel_size = [2,5], stride = [1,1], padding = 'same')	(1024, 9, 17)
ReLU_4	**(1024, 9, 17)**
Conv2d_5 (512, kernel_size = [2,5], stride = [1,1], padding = 'same')	(512, 9, 17)
ReLU_5	**(512, 9, 17)**
AvgPool_1 (kernel_size = [2,5] , stride = [2,2], padding = 'same')	**(512, 5, 9)**
Linear_1	(4096)
ReLU_6	**(4096)**
Dropout_1 (p=0.5)	**(4096)**
Linear_2	(num_classes)

where num_classes corresponds to the number of logits used for training each task (Table 1).

Thus, for CochCNN9, we extracted model representations from the following 10 layers with the number of unit activations (regressors) for each sound denoted in parentheses:
Cochleagram (211), ReLU_1 (6816), MaxPool2d_1 (3456), ReLU_2 (4608), MaxPool2d_2 (2304), ReLU_3 (4608), ReLU_4 (9216), ReLU_5 (4608), AvgPool_1 (2560), ReLU_6 (4096).

CochResNet50 architecture

The CochResNet50 model is composed of a ResNet50 backbone architecture applied to a cochleagram representation (with 2D convolutions applied to the cochleagram). Along with CochCNN9, this architecture was used for task-optimization comparisons throughout the paper.

The CochResNet50 architecture is denoted below with the sizes of the tensors propagated through the network denoted in parentheses. Model stages that were used for voxel and component response modeling are denoted in bold.

The model architecture follows:

Input (cochleagram)	**(1,211,390)**
Conv2d_1(1, 64, kernel_size=7, stride=2, padding=3, bias=False)	(64, 106, 195)
BatchNorm2d_1(64)	(64, 106, 195)
ReLU_1	**(64, 106, 195)**
MaxPool2d_1(kernel_size=3, stride=2, padding=1)	**(64, 53, 98)**
ResNetBlock_1(inplanes=64, planes=64, num_blocks=3, stride=1)	**(256, 53, 98)**
ResNetBlock_2(inplanes=256, planes=128, num_blocks=4, stride=2)	**(512, 27, 49)**
ResNetBlock_3(inplanes=512, planes=256, num_blocks=6, stride=2)	**(1024, 14, 25)**
ResNetBlock_4(inplanes=1024, planes=512, num_blocks=3, stride=2)	**(2048, 7, 13)**
AvgPool_1	(2048, 1, 1)
Linear_1	(num_classes)

where num_classes corresponds to the number of logits used for training each task (Table 1) and the ResNetBlock components of the architecture have the following structure:

1. input (x)
2. 1x1 Conv2d(inplanes, planes, stride=1)
3. BatchNorm2d(planes)
4. ReLU
5. 3x3 Conv2d (planes, planes, stride=1)
6. BatchNorm2d(planes)
7. ReLU
8. 1x1 Conv2d (planes, planes * expansion, stride=1)
9. BatchNorm2d(planes)
10. Residual connection on x (if inplanes !=planes * expansion): 1x1 Conv2D (inplanes, planes * expansion, stride)
11. Residual connection on x (if inplanes !=planes * expansion):

BatchNorm2d(planes * expansion)
12. Add output from (9) to output from (11)
13. (Output) ReLU

Multiple of these residual blocks (num_blocks) are stacked together to form a single ResNetBlock. The expansion factor was set to four for all layers (expansion=4).

Thus, for CochResNet50, we extracted model representations from the following 8 layers with the number of unit activations (regressors) for each sound denoted in parentheses:
Cochleagram (211), ReLU_1 (6784), MaxPool_1 (3392), ResNetBlock_1 (13568), ResNetBlock_2 (13824), ResNetBlock_3 (14336), ResNetBlock_4 (14336), AvgPool_1 (2048).

Training data set for CochResNet50 and CochCNN9 models - Word-Speaker-Noise tasks

Eight in-house models were trained on the Word-Speaker-Noise (WSN) data set. This data set was first presented in Feather et al., (2019)[36] and was constructed from existing speech recognition and environmental sound classification data sets. The data set description that follows is reproduced from [68] with some additions to further detail the speaker and environmental sound recognition tasks.

The data set was approximately balanced to enable performance of three tasks on the same training exemplar: (1) recognition of the word at the center of a two second speech clip (2) recognition of the speaker and (3) recognition of environmental sounds, that were superimposed with the speech clips (serving as “background noise” for the speech tasks while enabling an environmental sound recognition task).

The speech clips used in the data set were excerpted from the Wall Street Journal[120] (WSJ) and Spoken Wikipedia Corpora[121] (SWC). To choose speech clips, we screened WSJ, TIMIT[122] and a subset of articles from SWC for appropriate audio clips (specifically, clips that contained a word at least four characters long and that had one second of audio before the beginning of the word and after the end of the word, to enable the temporal jittering augmentation described below). Some SWC articles were left out of the screen due to a) potentially offensive content for human listening experiments; (29/1340 clips), b) missing data; (35/1340 clips), or c) bad audio quality (for example, due to computer generated voices of speakers reading the article or the talker changing mid-way through the clip; 33/1340 clips). Each segment was assigned the word class label of the word overlapping the segment midpoint and a speaker class label determined by the speaker. With the goal of constructing a data set with speaker and word class labels that were approximately independent, we selected words and speaker classes such that the exemplars from each class spanned at least 50 unique cross-class labels (e.g., 50 unique speakers for each of the word classes). This exclusion fully removed TIMIT from the training data set. We then selected words and speaker classes that each contained at least 200 unique utterances, and such that each class could contain a maximum of 25% of a single cross-class label (e.g., for a given word class, a maximum of 25% of utterances could come from the same speaker). These exemplars were subsampled so that the maximum number in any word or speaker class was less

than 2000. The resulting training data set contained 230,356 unique clips in 793 word classes and 432 speaker classes, with 40,650 unique clips in the test set. Each word class had between 200 and 2000 unique exemplars. A “null” class was used as a label for the word and speaker when a background clip was presented without the added speech.

The environmental soundtrack clips that were superimposed on the speech clips were a subset of examples from the “Unbalanced Train” split of the AudioSet data set (a set of annotated YouTube video soundtracks) [37] To minimize ambiguity for the two speech tasks, we removed any sounds under the “Speech” or “Whispering” branch of the AudioSet ontology. Since a high proportion of AudioSet clips contain music, we achieved a more balanced set by excluding any clips that were only labeled with the root label of “Music”, with no specific branch labels. We also removed silent clips by first discarding everything tagged with a “Silence” label and then culling clips containing more than 10% zeros. This screening resulted in a training set of 718,625 unique natural sound clips spanning 516 categories. Each AudioSet clip was a maximum of 10 seconds long, from which a 2-second excerpt was randomly cropped during training (see below). A “null” environmental sound label was used as a label when speech clips were presented without added background sound.

During training, the speech clips from the Word-Speaker-Noise data set were randomly cropped in time and superimposed on random crops of the AudioSet clips. Data augmentations during training consisted of 1) randomly selecting a clip from the pre-screened AudioSet clips to pair with each labeled speech clip, 2) randomly cropping 2 seconds of the AudioSet clip and 2 seconds of the speech clip, cropped such that the labeled word remained in the center of the clip (due to training pipeline technicalities, we used a pre-selected set of 5,810,600 paired speech and natural sound crops which spanned 25 epochs of the full set of speech clips and 8 passes through the full set of AudioSet clips), 3) superimposing the speech and the noise (i.e., the AudioSet crop) with a Signal-to-Noise-Ratio (SNR) sampled from a uniform distribution between -10dB SNR and 10dB SNR, augmented with additional samples of speech without an AudioSet background (i.e., with infinite SNR, 2464 examples in each epoch) and samples of AudioSet without speech (i.e., with negative infinite SNR, 2068 examples in each epoch) and 4) setting the root-mean-square (RMS) amplitude of the resulting signal to 0.1. By constructing the data set in this way, we could train networks on different tasks while using the same data set and training and test augmentations.

Evaluation performance for the word and speaker recognition tasks was measured from one pass through the speech test set (i.e., one crop from each of the 40,650 unique test set speech clips) constructed with the same augmentations used during training (specifically, variable SNR and temporal crops, paired with a set of AudioSet test clips from the “Balanced Train” split, same random seed used to test each model such that test sets were identical across models).

The representation from the AudioSet-trained models were evaluated with a support vector machine (SVM) fit to the ESC-50 data set[123], composed of 50 types of environmental sounds. After the model was trained, and for each of the five folds in ESC-50, an SVM was fit to the output representation of the top of the layer immediately before the final linear layer (AvgPool_1 for

CochResNet50 and ReLU_6 for CochCNN9). Each fold had 400 sounds, resulting in 1600 sounds used for training when holding out each fold. As the networks were trained with two-second-long sound clips, we took random two-second crops of the ESC-50 sounds. For each sound in the training and test data, we took 5 two-second-long crops at random from the five second sound (randomly selecting a new crop if the chosen crop was all zeros). The five crops of the training data were all used in fitting the SVM, treated as separate training data points. After the predictions were measured for the five crops for each test sound, we chose the label that was predicted most often as the prediction for the test sound.

The SVM was implemented with sklearn's LinearSVC, with cross validation over five regularization parameters (C=[0.01, 0.1, 1.0, 10.0, 100.0]). For cross validation, a random selection of 25% of the training sounds were held out and the SVM was fit on the other 75% of the sounds, and this was repeated three times (the five crops from a given sound were never split up between cross validation training and test splits, such that the cross validation tested for generalization to held-out sounds). This cross-validation strategy is independent of the held-out test fold, as it only relies on the training data set. A best regularization parameter was determined by choosing the parameter that resulted in the maximum percent correct averaged across the three splits, and we refit the SVM using the selected regularization parameter on the entire training data set of 1600 sounds to measure the performance on the held-out fold (400 sounds). The reported performance is the average across the 5 folds of the ESC-50 data set.

Training CochResNet50 and CochCNN9 models - Word-Speaker-Noise tasks

Each audio model was trained for 150 epochs of the speech data set (corresponding to 48 epochs of the AudioSet training data). The learning rate was decreased by a factor of 10 after every 50 speech epochs (16 AudioSet epochs). All models were trained on the OpenMind computing cluster at MIT using NVIDIA GPUs.

The Word and Speaker networks were trained with a cross entropy loss on the target labels. Because the AudioSet data set has multiple labels per clip, the logits are passed through a sigmoid and the Binary Cross Entropy is used as the loss function. Models had weight decay of 1e-4, except for models trained on the AudioSet task (including the multi-task models) which had weight decay of 0.

Both of the CochResNet50 and CochCNN9 architectures were trained simultaneously on all three tasks by including three fully connected layers as the final readout. These models were optimized by adding together a weighted loss from each individual task, and minimizing this summed loss. The weights used for the loss function were 1.0 (Word), 0.25 (Speaker), and 300 (AudioSet).

Additional training details are given in the table below.

Model Name	Batch Size	Initial Learning Rate	Num Classes (includes "null")	Accuracy on Training Task

CochCNN9 Word	128	0.01	794	(Top 1) 66.640% (Top 5) 83.102%
CochCNN9 Speaker	128	0.01	433	(Top 1) 96.216% (Top 5) 99.058%
CochCNN9 AudioSet	128	0.00001*	517	(ESC-50 SVM) 83.60%
CochCNN9 Multi-task	128	0.00001*	Three tasks: (Word) 794, (Speaker) 433, (AudioSet) 517	(Top 1 Word) 64.954% (Top 5 Word) 81.998% (Top 1 Speaker) 86.686% (Top 5 Speaker) 96.039% (ESC-50 SVM) 82.60%
CochResNet50 Word	256	0.1	794	(Top 1) 86.792% (Top 5) 95.149%
CochResNet50 Speaker	256	0.1	433	(Top 1) 99.114% (Top 5) 99.835%
CochResNet50 AudioSet	256	0.001*	517	(ESC-50 SVM) 91.6%
CochResNet50 Multi-task	256	0.001*	Three tasks: (Word) 794, (Speaker) 433, (AudioSet) 517	(Top 1 Word) 83.459% (Top 5 Word) 93.422% (Top 1 Speaker) 94.354% (Top 5 Speaker) 98.785% (ESC-50 SVM) 87.450%

* Models trained with the AudioSet loss had additional gradient clipping (max l_2 norm=1.0) and learning rate warm-up for the first 500 batches of training (learning rate = <initial learning rate> / (500-i), where i is the batch number).

Training data set for CochResNet50 and CochCNN9 models - musical genre task

The genre task was the 41-way classification task introduced by Kell et al., (2018). The sounds and labels were derived from The Million Song Dataset[124]. Genre labels were obtained from user-generated ‘‘tags’’ from the MusicBrainz open-source music encyclopedia (https://musicbrainz.org/). Tags were first culled to eliminate those that did not apply to at least ten different artists or that did not obviously correspond to a genre. These tags were then grouped into genre classes using hierarchical clustering applied to the tag co-occurrence matrix, grouping together tags that overlapped substantially. See Table S2 from the Kell et al., (2018) for a list of genres and the tags associated with each genre.

Training exemplars for the genre task were obtained by randomly excerpting two-second clips from the tracks that had tags for the genre labels selected for the task. The music excerpts were superimposed with two-second excerpts of one of four different background noises: (1) auditory scenes, (2) two-speaker speech babble, (3) eight-speaker speech babble, or (4) music-shaped noise. Music-shaped noise consisted of a two-second clip of noise that was matched to the average spectrum of its corresponding two-second clip of music. SNRs were selected to yield performance in human listeners that was below ceiling (but above chance). The mean SNR for each of the four background types was 12 dB, with the SNR for each training example drawn randomly from a Gaussian with a standard deviation of 2 dB. All waveforms were downsampled to 16 kHz.

Training CochResNet50 and CochCNN9 models - musical genre task

The genre networks were trained with a cross entropy loss, and. A stochastic gradient descent optimizer was used for training with weight decay of 1e-4, momentum of 0.9, and an initial learning rate of 0.01. The models were trained for 125 epochs of the genre data set, and the learning rate was dropped by a factor of 10 after every 50 epochs. A batch size of 64 was used for training. The CochCNN9 architecture achieved Top 1 accuracy of 83.21% and Top 5 of 96.19% on the musical genre task, and the CochResNet50 model achieved Top 1 accuracy of 87.99% and Top 5 accuracy of 97.56%.

Candidate models with permuted weights

In addition to the trained networks, we also analyzed ‘permuted’ versions of the models with the exact same architecture as the trained models. We created these models by replacing all parameters making up the trained model in each network by random permutations across all tensor dimensions within a given parameter block (e.g. a weight or bias matrix) for each model stage. This model manipulation destroyed the parameter structure learned during task-optimization, while preserving the marginal statistics of the parameters. All analyses procedures were identical for trained and permuted networks.

Acknowledgements

We thank Ian Griffith for training the music genre classification models, Alex Kell for helpful discussions, Nancy Kanwisher for sharing fMRI data, developers for making their trained models available for public use, and the McDermott lab for comments on an earlier draft of the paper. Work supported by NIH grant R01DC017970. G.T. was supported by the Amazon Fellowship from the Science Hub (administered by the MIT Schwarzman College of Computing) and the International Doctoral Fellowship from American Association of University Women (AAUW). J.F was supported by an MIT Friends of McGovern Institute Fellowship and a DOE Computational Science Graduate Fellowship (grant DE-FG02-97ER25308).

References

1. Marblestone, A. H., Wayne, G. & Kording, K. P. Toward an integration of deep learning and neuroscience. *Front. Comput. Neurosci.* **10**, (2016).
2. Richards, B. A. *et al.* A deep learning framework for neuroscience. *Nat. Neurosci.* **22**, 1761–1770 (2019).
3. Storrs, K. R. & Kriegeskorte, N. Deep learning for cognitive neuroscience. *ArXiv190301458 Cs Q-Bio* (2019).
4. Kell, A. J. E. & McDermott, J. H. Deep neural network models of sensory systems: windows onto the role of task constraints. *Curr. Opin. Neurobiol.* **55**, 121–132 (2019).
5. Schrimpf, M. *et al.* Integrative benchmarking to advance neurally mechanistic models of human intelligence. *Neuron* **108**, 413–423 (2020).
6. Saxe, A., Nelli, S. & Summerfield, C. If deep learning is the answer, what is the question? *Nat. Rev. Neurosci.* **22**, 55–67 (2021).
7. Lake, B. M., Zaremba, W., Fergus, R. & Gureckis, T. M. Deep neural networks predict category typicality ratings for images. *Cogn. Sci.* 6 (2015).
8. Peterson, J. C., Abbott, J. T. & Griffiths, T. L. Evaluating (and improving) the correspondence between deep neural networks and human representations. *Cogn. Sci.* **42**, 2648–2669 (2018).
9. King, M. L., Groen, I. I. A., Steel, A., Kravitz, D. J. & Baker, C. I. Similarity judgments and cortical visual responses reflect different properties of object and scene categories in naturalistic images. *NeuroImage* **197**, 368–382 (2019).
10. Jang, H., McCormack, D. & Tong, F. Noise-trained deep neural networks effectively predict human vision and its neural responses to challenging images. *PLOS Biol.* **19**, e3001418 (2021).
11. Yamins, D. L. K. *et al.* Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci.* **111**, 8619–8624 (2014).
12. Guclu, U. & van Gerven, M. A. J. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* **35**, 10005–10014 (2015).
13. Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A. & Oliva, A. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Sci. Rep.* **6**, 27755 (2016).
14. Eickenberg, M., Gramfort, A., Varoquaux, G. & Thirion, B. Seeing it all: Convolutional network layers map the function of the human visual system. *NeuroImage* **152**, 184–194 (2017).
15. Kell, A. J. E., Yamins, D. L. K., Shook, E. N., Norman-Haignere, S. V. & McDermott, J. H. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron* **98**, 630-644.e16 (2018).
16. Saddler, M. R., Gonzalez, R. & McDermott, J. H. Deep neural network models reveal interplay of peripheral coding and stimulus statistics in pitch perception. *Nat. Commun.* **12**, 7278 (2021).
17. Francl, A. & McDermott, J. H. Deep neural network models of sound localization reveal how perception is adapted to real-world environments. *Nat. Hum. Behav.* **6**, 111–133 (2022).
18. Brochier, T. *et al.* From microphone to phoneme: an end-to-end computational neural model for predicting speech perception with cochlear implants. *IEEE Trans. Biomed. Eng.* **PP**, (2022).
19. Millet, J. & Dunbar, E. Do self-supervised speech models develop human-like perception biases? in *Proceedings of the 60th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers) 7591–7605 (Association for Computational Linguistics, 2022). doi:10.18653/v1/2022.acl-long.523.
20. Güçlü, U., Thielen, J., Hanke, M. & van Gerven, M. A. J. Brains on beats. *ArXiv160602627 Q-Bio* (2016).
21. Koumura, T., Terashima, H. & Furukawa, S. Cascaded tuning to amplitude modulation for natural sound recognition. *J. Neurosci.* **39**, 5517–5533 (2019).
22. Khatami, F. & Escabí, M. A. Spiking network optimized for word recognition in noise predicts auditory system hierarchy. *PLOS Comput. Biol.* **16**, e1007558 (2020).
23. Magnuson, J. S. *et al.* EARSHOT: a minimal neural network model of incremental human speech recognition. *Cogn. Sci.* **44**, (2020).
24. Millet, J. & King, J.-R. Inductive biases, pretraining and fine-tuning jointly account for brain responses to speech. *ArXiv210301032 Cs Eess Q-Bio* (2021).
25. Millet, J. *et al.* Toward a realistic model of speech processing in the brain with self-supervised learning. Preprint at http://arxiv.org/abs/2206.01685 (2022).
26. Li, Y. *et al.* Dissecting neural computations of the human auditory pathway using deep neural networks for speech. *bioRxiv* (2022).
27. Vaidya, A. R., Jain, S. & Huth, A. G. Self-supervised models of audio effectively explain human cortical responses to speech. (2022) doi:10.48550/ARXIV.2205.14252.
28. Khaligh-Razavi, S.-M. & Kriegeskorte, N. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS Comput. Biol.* **10**, e1003915 (2014).
29. Schrimpf, M. *et al.* The neural architecture of language: Integrative modeling converges on predictive processing. *Proc. Natl. Acad. Sci.* **118**, e2105646118 (2021).
30. Caucheteux, C. & King, J.-R. Brains and algorithms partially converge in natural language processing. *Commun. Biol.* **5**, 134 (2022).
31. Kriegeskorte, N., Mur, M. & Bandettini, P. Representational similarity analysis - connecting the branches of systems neuroscience. *Front. Syst. Neurosci.* **2**, (2008).
32. Chi, T., Ru, P. & Shamma, S. A. Multiresolution spectrotemporal analysis of complex sounds. *J. Acoust. Soc. Am.* **118**, 887–906 (2005).
33. Norman-Haignere, S. V., Kanwisher, N. G. & McDermott, J. H. Distinct cortical pathways for music and speech revealed by hypothesis-free voxel decomposition. *Neuron* **88**, 1281–1296 (2015).
34. Boebinger, D., Norman-Haignere, S. V., McDermott, J. H. & Kanwisher, N. Music-selective neural populations arise without musical training. *J. Neurophysiol.* **125**, 2237–2263 (2021).
35. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *ArXiv151203385 Cs* (2015).
36. Feather, J., Durango, A., Gonzalez, R. & McDermott, J. H. Metamers of neural networks reveal divergence from human perceptual systems. in *Advances in Neural Information Processing Systems* vol. 32 (Curran Associates, Inc., 2019).
37. Gemmeke, J. F. *et al.* Audio Set: An ontology and human-labeled dataset for audio events. in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 776–780 (2017). doi:10.1109/ICASSP.2017.7952261.
38. Naselaris, T., Kay, K. N., Nishimoto, S. & Gallant, J. L. Encoding and decoding in fMRI. *NeuroImage* **56**, 400–410 (2011).
39. Santoro, R. *et al.* Encoding of natural sounds at multiple spectral and temporal resolutions in the human auditory cortex. *PLOS Comput. Biol.* **10**, e1003412 (2014).
40. Huth, A. G., de Heer, W. A., Griffiths, T. L., Theunissen, F. E. & Gallant, J. L. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature* **532**, 453–458 (2016).
41. Heer, W. A. de, Huth, A. G., Griffiths, T. L., Gallant, J. L. & Theunissen, F. E. The hierarchical cortical organization of human speech processing. *J. Neurosci.* **37**, 6539–6557 (2017).

42. Pereira, F. *et al.* Toward a universal decoder of linguistic meaning from brain activation. *Nat. Commun.* **9**, 963 (2018).
43. Spearman, C. The proof and measurement of association between two things. *Am. J. Psychol.* **15**, 72 (1904).
44. Schoppe, O., Harper, N. S., Willmore, B. D. B., King, A. J. & Schnupp, J. W. H. Measuring the performance of neural models. *Front. Comput. Neurosci.* **10**, (2016).
45. Storrs, K. R., Kietzmann, T. C., Walther, A., Mehrer, J. & Kriegeskorte, N. Diverse deep neural networks all predict human inferior temporal cortex well, after training and fitting. *J. Cogn. Neurosci.* 1–21 (2021) doi:10.1162/jocn_a_01755.
46. Xu, Y. & Vaziri-Pashkam, M. Limits to visual representational correspondence between convolutional neural networks and the human brain. *Nat. Commun.* **12**, 2065 (2021).
47. Glasser, M. F. *et al.* A multi-modal parcellation of human cerebral cortex. *Nature* **536**, 171–178 (2016).
48. Elmoznino, E. & Bonner, M. F. High-performing neural network models of visual cortex benefit from high latent dimensionality. 2022.07.13.499969 Preprint at https://doi.org/10.1101/2022.07.13.499969 (2022).
49. Wessinger, C. M. *et al.* Hierarchical organization of the human auditory cortex revealed by functional magnetic resonance imaging. *J. Cogn. Neurosci.* **13**, 1–7 (2001).
50. Rauschecker, J. P. & Scott, S. K. Maps and streams in the auditory cortex: nonhuman primates illuminate human speech processing. *Nat. Neurosci.* **12**, 718–724 (2009).
51. Okada, K. *et al.* Hierarchical organization of human auditory cortex: evidence from acoustic invariance in the response to intelligible speech. *Cereb. Cortex N. Y. N 1991* **20**, 2486–2495 (2010).
52. Liebenthal, E., Binder, J. R., Spitzer, S. M., Possing, E. T. & Medler, D. A. Neural substrates of phonemic perception. *Cereb. Cortex N. Y. N 1991* **15**, 1621–1631 (2005).
53. Uppenkamp, S., Johnsrude, I. S., Norris, D., Marslen-Wilson, W. & Patterson, R. D. Locating the initial stages of speech-sound processing in human temporal cortex. *NeuroImage* **31**, 1284–1296 (2006).
54. Chang, E. F. *et al.* Categorical speech representation in human superior temporal gyrus. *Nat. Neurosci.* **13**, 1428–1432 (2010).
55. Peelle, J. E., Johnsrude, I. S. & Davis, M. H. Hierarchical processing for speech in human auditory cortex and beyond. *Front. Hum. Neurosci.* **4**, 51 (2010).
56. Obleser, J., Leaver, A., VanMeter, J. & Rauschecker, J. Segregation of vowels and consonants in human auditory cortex: evidence for distributed hierarchical organization. *Front. Psychol.* **1**, (2010).
57. Overath, T., McDermott, J. H., Zarate, J. M. & Poeppel, D. The cortical analysis of speech-specific temporal structure revealed by responses to sound quilts. *Nat. Neurosci.* **18**, 903–911 (2015).
58. Evans, S. & Davis, M. H. Hierarchical organization of auditory and motor representations in speech perception: evidence from searchlight similarity analysis. *Cereb. Cortex* **25**, 4772–4788 (2015).
59. Norman-Haignere, S. V. *et al.* A neural population selective for song in human auditory cortex. *Curr. Biol.* **32**, 1470-1484.e12 (2022).
60. Norman-Haignere, S. V. *et al.* Multiscale temporal integration organizes hierarchical computation in human auditory cortex. *Nat. Hum. Behav.* **6**, 455–469 (2022).
61. Norman-Haignere, S. V. & McDermott, J. H. Neural responses to natural and model-matched stimuli reveal distinct computations in primary and nonprimary auditory cortex. *PLOS Biol.* **16**, e2005127 (2018).
62. Kell, A. J. E. & McDermott, J. H. Invariance to background noise as a signature of non-primary auditory cortex. *Nat. Commun.* **10**, 3958 (2019).

63. Hamilton, L. S., Oganian, Y., Hall, J. & Chang, E. F. Parallel and distributed encoding of speech across human auditory cortex. *Cell* **184**, 4626-4639.e13 (2021).
64. Cox, D. D. & Savoy, R. L. Functional magnetic resonance imaging (fMRI) 'brain reading': detecting and classifying distributed patterns of fmri activity in human visual cortex. *NeuroImage* **19**, 261–270 (2003).
65. Ivanova, A. A. *et al.* Beyond linear regression: mapping models in cognitive neuroscience should align with research goals. Preprint at https://doi.org/10.48550/arXiv.2208.10668 (2022).
66. Szegedy, C. *et al.* Intriguing properties of neural networks. (2013) doi:10.48550/ARXIV.1312.6199.
67. Nguyen, A., Yosinski, J. & Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 427–436 (2015). doi:10.1109/CVPR.2015.7298640.
68. Feather, J., Leclerc, G., Mądry, A. & McDermott, J. H. Model metamers illuminate divergences between biological and artificial neural networks. 2022.05.19.492678 Preprint at https://doi.org/10.1101/2022.05.19.492678 (2022).
69. Ratan Murty, N. A., Bashivan, P., Abate, A., DiCarlo, J. J. & Kanwisher, N. Computational models of category-selective brain regions enable high-throughput tests of selectivity. *Nat. Commun.* **12**, 5540 (2021).
70. Hickok, G. & Poeppel, D. The cortical organization of speech processing. *Nat. Rev. Neurosci.* **8**, 393–402 (2007).
71. Rauschecker, J. P. & Tian, B. Mechanisms and streams for processing of "what" and "where" in auditory cortex. *Proc. Natl. Acad. Sci.* **97**, 11800–11806 (2000).
72. Alain, C., Arnott, S. R., Hevenor, S., Graham, S. & Grady, C. L. "What" and "where" in the human auditory system. *Proc. Natl. Acad. Sci.* **98**, 12301–12306 (2001).
73. Warren, J. D. & Griffiths, T. D. Distinct mechanisms for processing spatial sequences and pitch sequences in the human auditory brain. *J. Neurosci.* **23**, 5799–5804 (2003).
74. Ahveninen, J. *et al.* Task-modulated "what" and "where" pathways in human auditory cortex. *Proc. Natl. Acad. Sci. U. S. A.* **103**, 14608–14613 (2006).
75. Lomber, S. G. & Malhotra, S. Double dissociation of 'what' and 'where' processing in auditory cortex. *Nat. Neurosci.* **11**, 609–616 (2008).
76. Bizley, J. K. & Cohen, Y. E. The what, where and how of auditory-object perception. *Nat. Rev. Neurosci.* **14**, 693–707 (2013).
77. Hamilton, L. S., Edwards, E. & Chang, E. F. A spatial map of onset and sustained responses to speech in the human superior temporal gyrus. *Curr. Biol.* **28**, 1860-1871.e4 (2018).
78. Forseth, K. J., Hickok, G., Rollo, P. S. & Tandon, N. Language prediction mechanisms in human auditory cortex. *Nat. Commun.* **11**, 5240 (2020).
79. Lindsay, G. W. Convolutional neural networks as a model of the visual system: past, present, and future. *J. Cogn. Neurosci.* **33**, 2017–2031 (2021).
80. Zhuang, C. *et al.* Unsupervised neural network models of the ventral visual stream. *Proc. Natl. Acad. Sci.* **118**, e2014196118 (2021).
81. Chen, H. *et al.* Unsupervised segmentation in real-world images via spelke object inference. Preprint at https://doi.org/10.48550/arXiv.2205.08515 (2022).
82. Bashivan, P., Kar, K. & DiCarlo, J. J. Neural population control via deep image synthesis. *Science* **364**, eaav9436 (2019).
83. Xiao, W. & Kreiman, G. XDream: Finding preferred stimuli for visual neurons using generative networks and gradient-free optimization. *PLOS Comput. Biol.* **16**, e1007973 (2020).

84. Sexton, N. J. & Love, B. C. Reassessing hierarchical correspondences between brain and deep networks through direct interface. *Sci. Adv.* **8**, eabm2219 (2022).
85. Keshishian, M. *et al.* Estimating and interpreting nonlinear receptive field of sensory neural responses with deep neural network models. *eLife* **9**, e53445 (2020).
86. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).
87. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
88. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
89. Del Giudice, M. Effective dimensionality: a tutorial. *Multivar. Behav. Res.* **56**, 527–542 (2021).
90. Jenkinson, M. & Smith, S. A global optimisation method for robust affine registration of brain images. *Med. Image Anal.* **5**, 143–156 (2001).
91. Greve, D. N. & Fischl, B. Accurate and robust brain image alignment using boundary-based registration. *NeuroImage* **48**, 63–72 (2009).
92. Dale, A. M., Fischl, B. & Sereno, M. I. Cortical surface-based analysis. I. Segmentation and surface reconstruction. *NeuroImage* **9**, 179–194 (1999).
93. Penhune, V. B. Sensitive periods in human development: evidence from musical training. *Cortex J. Devoted Study Nerv. Syst. Behav.* **47**, 1126–1137 (2011).
94. Gong, Y., Chung, Y.-A. & Glass, J. AST: Audio Spectrogram Transformer. Preprint at https://doi.org/10.48550/arXiv.2104.01778 (2021).
95. Deng, J. *et al.* ImageNet: A large-scale hierarchical image database. in *2009 IEEE Conference on Computer Vision and Pattern Recognition* 248–255 (2009). doi:10.1109/CVPR.2009.5206848.
96. Drossos, K., Adavanne, S. & Virtanen, T. Automated audio captioning with recurrent neural networks. in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)* 374–378 (2017). doi:10.1109/WASPAA.2017.8170058.
97. Drossos, K., Lipping, S. & Virtanen, T. Clotho: an audio captioning dataset. in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 736–740 (2020). doi:10.1109/ICASSP40776.2020.9052990.
98. Amodei, D. *et al.* Deep Speech 2: end-to-end speech recognition in english and mandarin. in *Proceedings of The 33rd International Conference on Machine Learning* 173–182 (PMLR, 2016).
99. Panayotov, V., Chen, G., Povey, D. & Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 5206–5210 (2015). doi:10.1109/ICASSP.2015.7178964.
100. Fu, S.-W. *et al.* MetricGAN+: An Improved Version of MetricGAN for Speech Enhancement. Preprint at https://doi.org/10.48550/arXiv.2104.03538 (2021).
101. Veaux, C., Yamagishi, J. & King, S. The voice bank corpus: Design, collection and data analysis of a large regional accent speech database. in *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)* 1–4 (2013). doi:10.1109/ICSDA.2013.6709856.
102. Wang, C. *et al.* fairseq S2T: fast speech-to-text modeling with fairseq. in *Proceedings of the 2020 Conference of the Asian Chapter of the Association for Computational Linguistics (AACL): System Demonstrations* (2020).
103. Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M. & Zhong, J. Attention is all you need in speech separation. in (2021).

104. Maciejewski, M., Wichern, G., McQuinn, E. & Roux, J. L. WHAMR!: noisy and reverberant single-channel speech separation. Preprint at https://doi.org/10.48550/arXiv.1910.10279 (2020).
105. Hershey, S. *et al.* CNN architectures for large-scale audio classification. Preprint at https://doi.org/10.48550/arXiv.1609.09430 (2017).
106. van Niekerk, B., Nortje, L. & Kamper, H. Vector-quantized neural networks for acoustic unit discovery in the ZeroSpeech 2020 challenge. Preprint at https://doi.org/10.48550/arXiv.2005.09409 (2020).
107. Dunbar, E. *et al.* The Zero Resource Speech Challenge 2019: TTS without T. Preprint at https://doi.org/10.48550/arXiv.1904.11469 (2019).
108. Baevski, A., Zhou, H., Mohamed, A. & Auli, M. Wav2vec 2.0: a framework for self-supervised learning of speech representations. Preprint at https://doi.org/10.48550/arXiv.2006.11477 (2020).
109. Touvron, H. *et al.* Training data-efficient image transformers & distillation through attention. Preprint at https://doi.org/10.48550/arXiv.2012.12877 (2021).
110. Vaswani, A. *et al.* Attention Is All You Need. *ArXiv170603762 Cs* (2017).
111. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2018) doi:10.48550/arXiv.1810.04805.
112. Ravanelli, M. *et al.* SpeechBrain: a general-purpose speech toolkit. Preprint at https://doi.org/10.48550/arXiv.2106.04624 (2021).
113. Wolf, T. *et al.* Transformers: state-of-the-art natural language processing. in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* 38–45 (Association for Computational Linguistics, 2020). doi:10.18653/v1/2020.emnlp-demos.6.
114. Kudo, T. & Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. Preprint at https://doi.org/10.48550/arXiv.1808.06226 (2018).
115. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. Preprint at https://doi.org/10.48550/arXiv.1409.1556 (2015).
116. Dunbar, E. *et al.* The Zero Resource Speech Challenge 2020: Discovering discrete subword and word units. Preprint at https://doi.org/10.48550/arXiv.2010.05967 (2020).
117. Graves, A., Fernández, S. & Gomez, F. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. in *In Proceedings of the International Conference on Machine Learning, ICML 2006* 369–376 (2006).
118. Glasberg, B. R. & Moore, B. C. J. Derivation of auditory filter shapes from notched-noise data. *Hear. Res.* **47**, 103–138 (1990).
119. McDermott, J. H. & Simoncelli, E. P. Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis. *Neuron* **71**, 926–940 (2011).
120. Paul, D. B. & Baker, J. M. The design for the Wall Street Journal-based CSR corpus. in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992* (1992).
121. Köhn, A., Stegen, F. & Baumann, T. Mining the spoken wikipedia for speech data and beyond. in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* 4644–4647 (European Language Resources Association (ELRA), 2016).
122. Zue, V. & Seneff, S. Transcription and alignment of the TIMIT database. in (1996). doi:10.1016/B978-044481607-8/50088-8.
123. Piczak, K. J. ESC: dataset for environmental sound classification. in *Proceedings of the 23rd ACM international conference on Multimedia* 1015–1018 (Association for Computing Machinery, 2015). doi:10.1145/2733373.2806390.

124. Bertin-Mahieux, T., Whitman, B. & Lamere, P. The Million Song Dataset. in *In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)* (2011).

Supplemental Information

Many but not all deep neural network audio models capture brain responses and exhibit hierarchical region correspondence

Tuckute*, Feather*, Boebinger & McDermott (2022).

Supplementary Figure S1

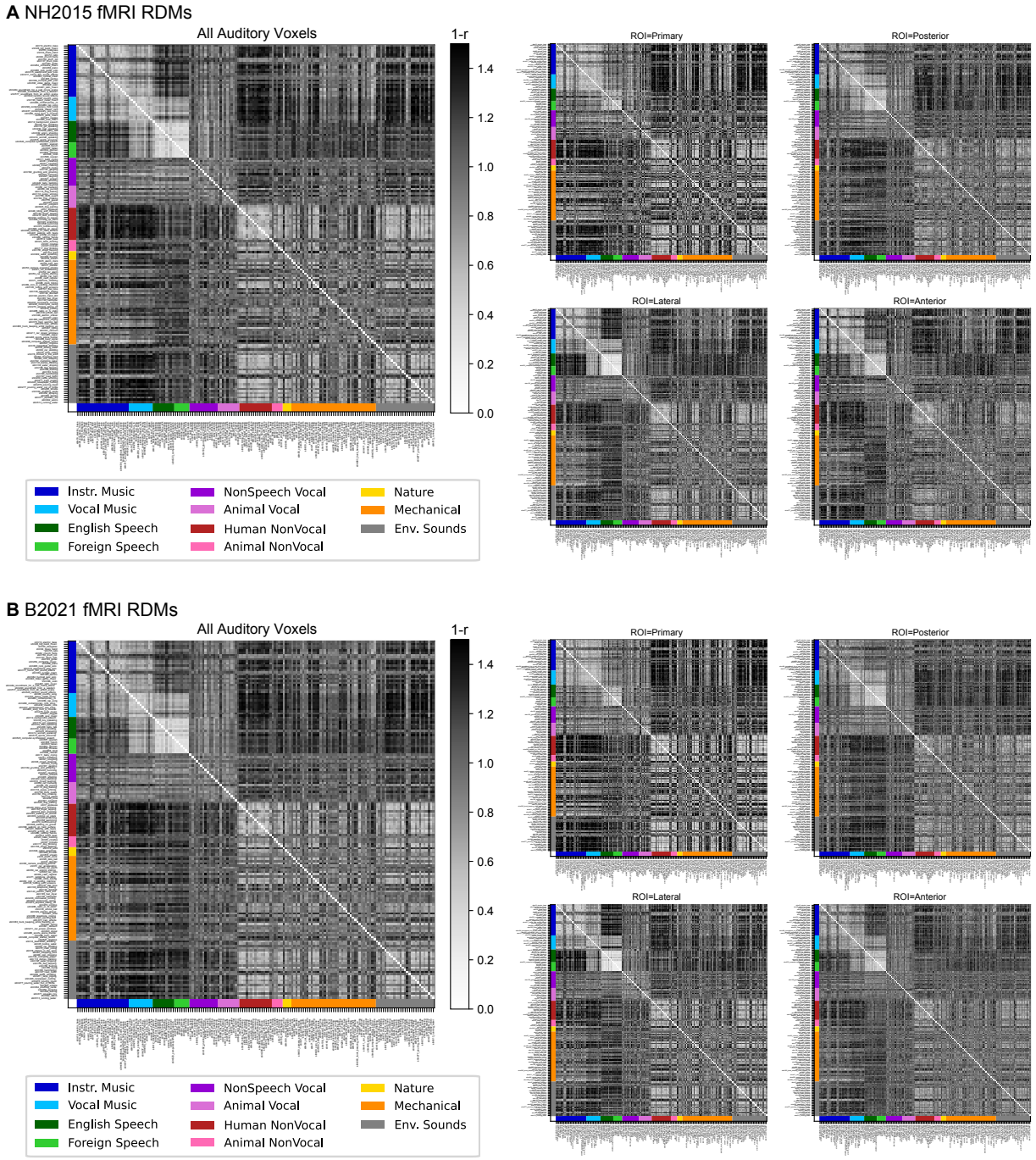
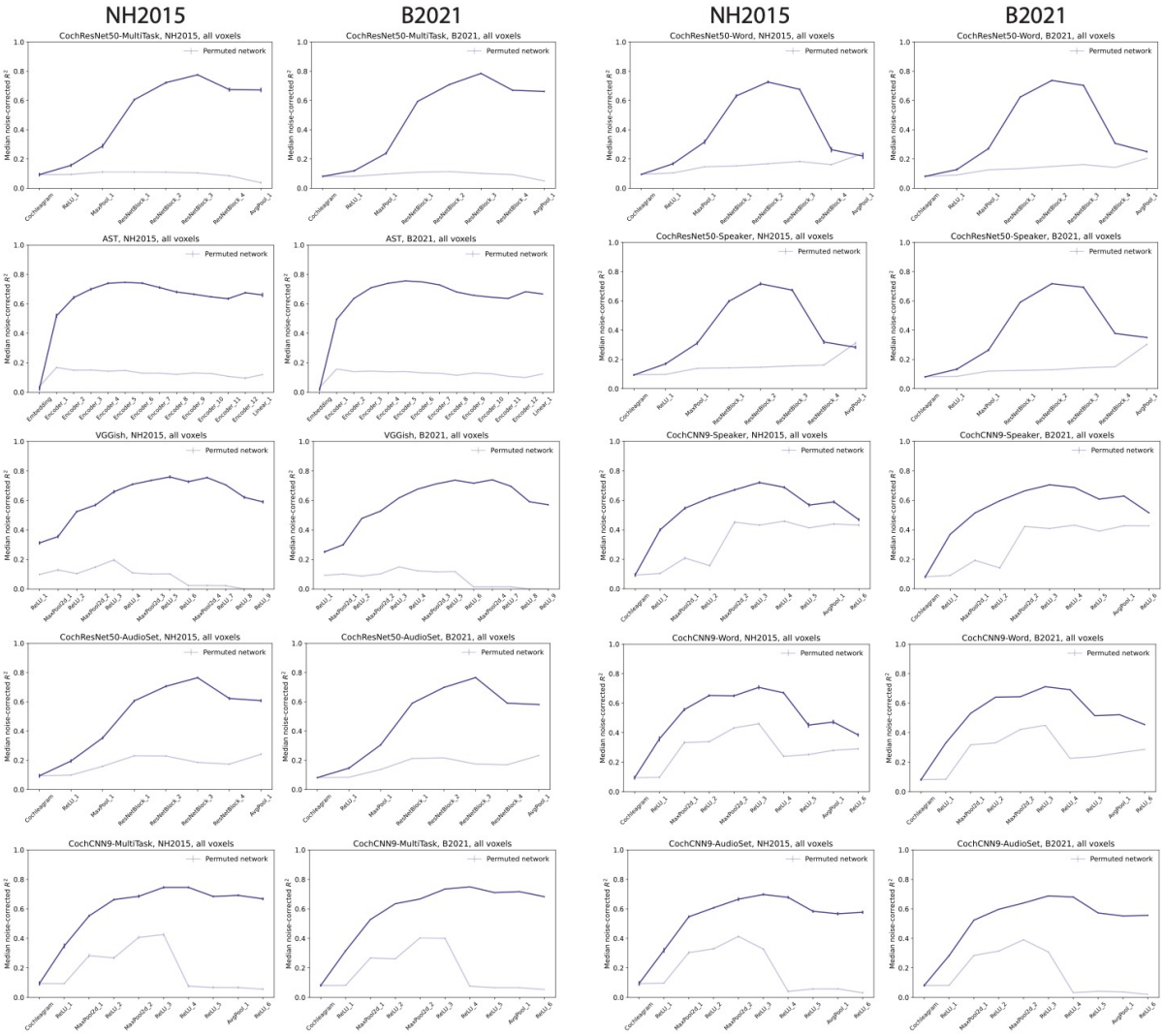


Figure S1. Representational Dissimilarity Matrices for fMRI voxels in (A) NH2015 and (B) B2021. For visualization purposes, the RDMs are computed as 1-Spearman Correlation between the 3-scan average activations for pairs of sounds. RDMs are computed for all sound-responsive voxels (left) and using only a subset of voxels for each of the anatomical ROIs (right). Sounds are grouped by sound categories (included in colors on the axis).

Supplementary Figure S2

Predictivity across Model Stages (subfigure 1/2)



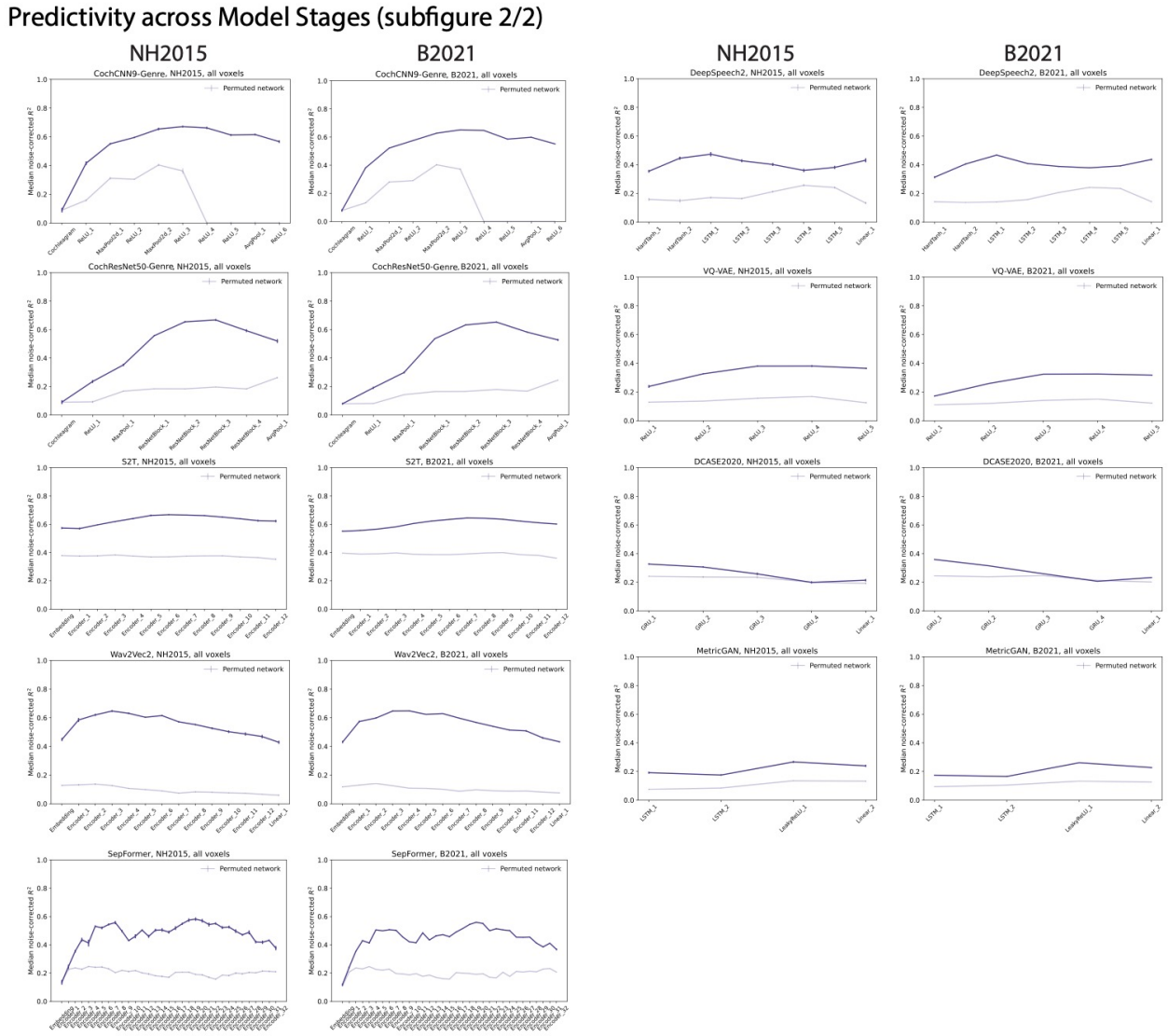


Figure S2. Median variance explained across model stages for each model. Explained variance was measured for each voxel and the aggregated median variance explained across all voxels in auditory cortex was obtained. This aggregated median variance explained is plotted for all candidate models (n=19) for both fMRI datasets. The model plots are sorted according to overall model performance (median noise-corrected R^2 for NH2015 in Figure 2Ai in the main text), meaning that the first subplot in subfigure 1 shows the best-performing model, CochResNet50-MultiTask, and the last subplot in subfigure 2 shows the worst-performing model, MetricGAN. Dark lines show the trained networks, and lighter lines show the control networks with permuted weights. Error bars are within-participant SEM. Error bars are smaller for the B2021 dataset because of the larger number of participants (20 vs. 8).

Supplementary Figure S3 (extension of Figure 6)

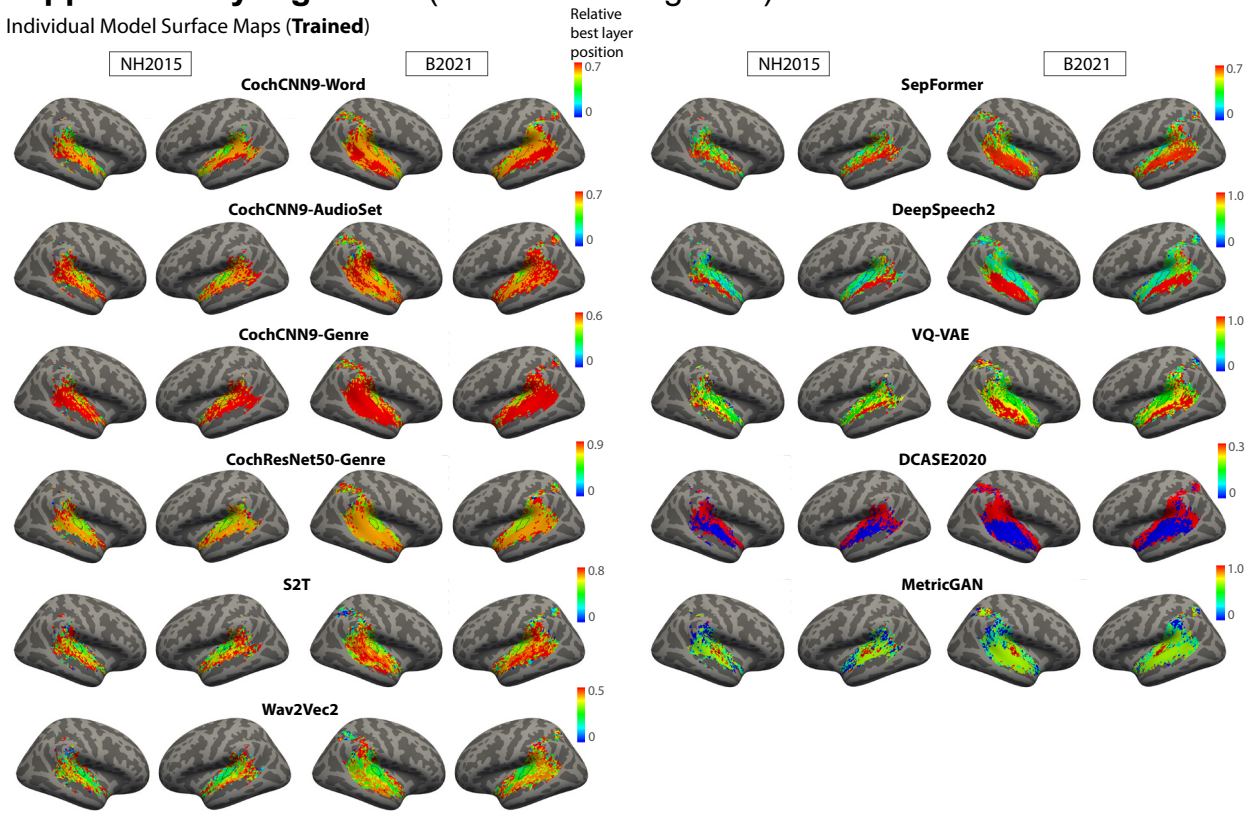


Figure S3. Surface maps of best-predicting model stage for trained models. The figure shows surface maps for trained models that are not included in Figure 6A in the main text (which featured the n=8 best-predicting models, leaving the n=11 models shown here). The plots are sorted according to overall model predictivity (the quantity plotted in Figure 2Ai in the main text). As in Figure 6A in the main text, the plots show the model stage that best predicts each voxel as a surface map (FsAverage) (median best stage across participants). We assigned each model stage a position index between 0 and 1. The color scale limits were set to extend from 0 to the stage beyond the most common best stage (across voxels).

Supplementary Figure S4 (extension of Figure 6)

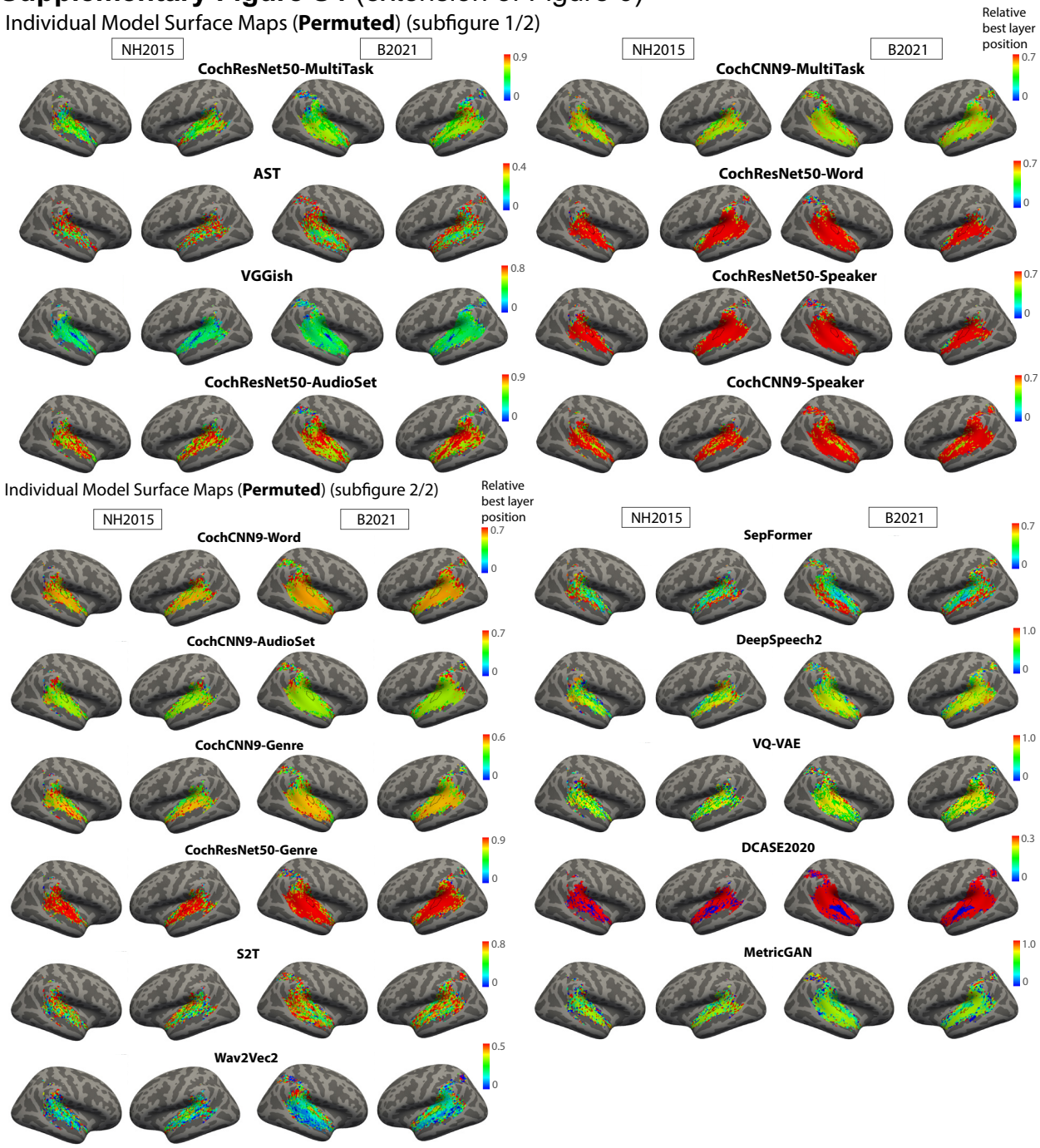


Figure S4. Surface maps of best-predicting model stage for permuted control models. Subfigure 1 shows the surface maps for the eight models shown in Figure 6A, but with permuted weights. Subfigure 2 shows surface maps for models with permuted weights that are not included in Figure 6A in the main text. Identical analyses procedures and color scale limits were used for the permuted models as for the trained ones.

Supplementary Figure S5 (extension of Figure 7)

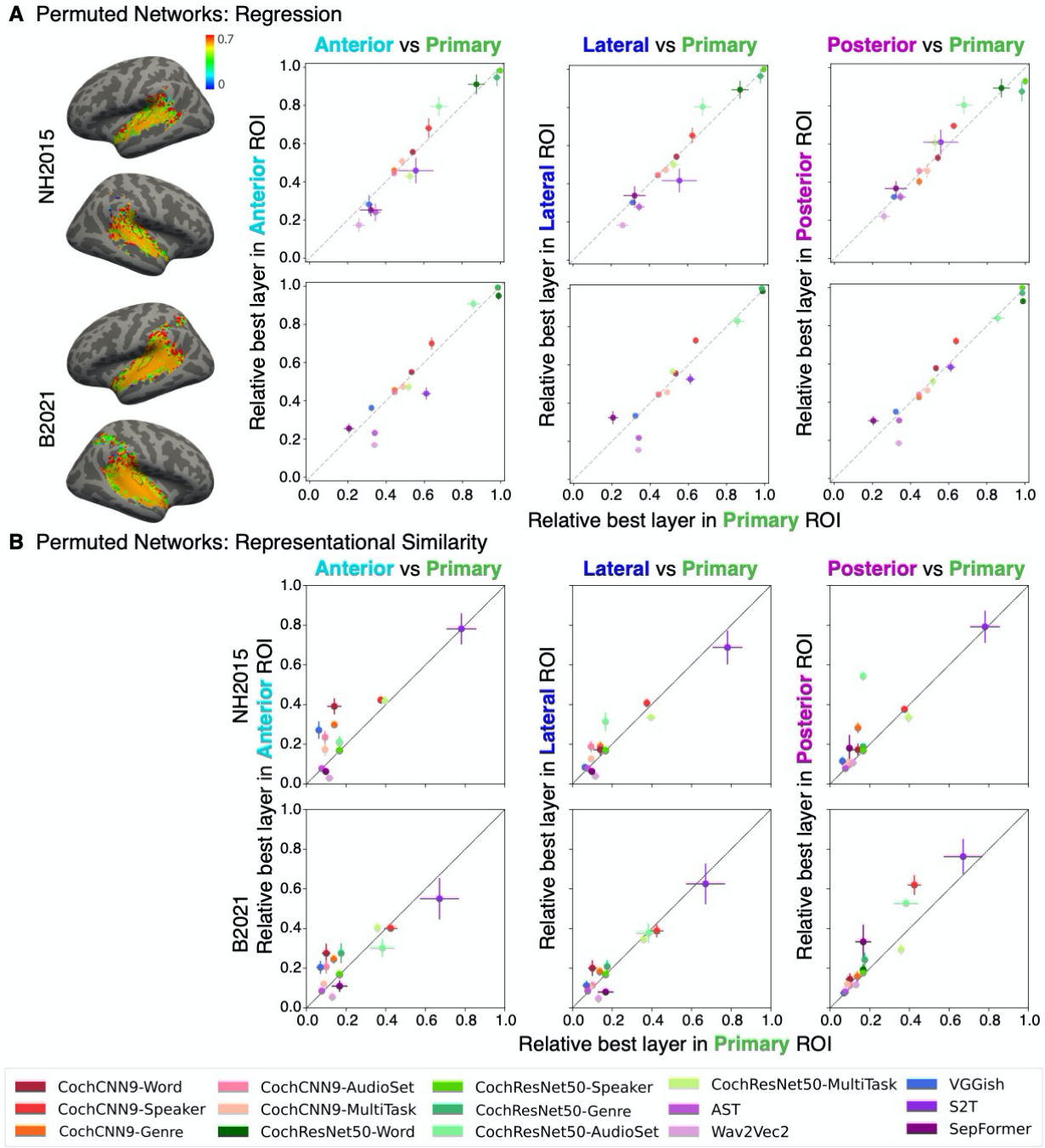


Figure S5. Stage-region correspondence of permuted networks. The figure mirrors Figure 7 in the main text which shows the quantification of model-stage-region correspondence across trained networks. Given that the first stage of the in-house models (cochleagram input stage) could not be permuted, this stage was excluded for these analyses for all the ten in-house models. **(A)** As in Figure 7 in the main text, we obtained the median best-predicting stage for each model within four anatomical ROIs (illustrated in Figure 7A, main text): primary auditory cortex (x axis in each plot in A and B) and anterior, lateral, and posterior non-primary regions (y axes in A and B). We performed the

analysis on each of the two fMRI data sets, including each model that out-predicted the baseline model in Figure 2Ai in the main text (n=15 models). Each data point corresponds to a model with permuted weights, with the same color correspondence as in Figure 2 in the main text. None of the six possible comparisons (two datasets x three non-primary ROIs) were statistically significant without correction for multiple comparisons, $p>0.33$ in all cases (Wilcoxon signed rank tests, two-tailed). **(B)** Same analysis as (A) but with the best-matching model stage determined by correlations between the model and ROI representational dissimilarity matrices. Three of the six possible comparisons were statistically significant without correction for multiple comparisons: $p=0.033$ NH2015 Primary vs. Anterior, $p=0.047$ NH2015 Primary vs. Posterior, $p=0.008$ B2021 Primary vs. Posterior (Wilcoxon signed rank tests, two-tailed).

Supplementary Figure S6

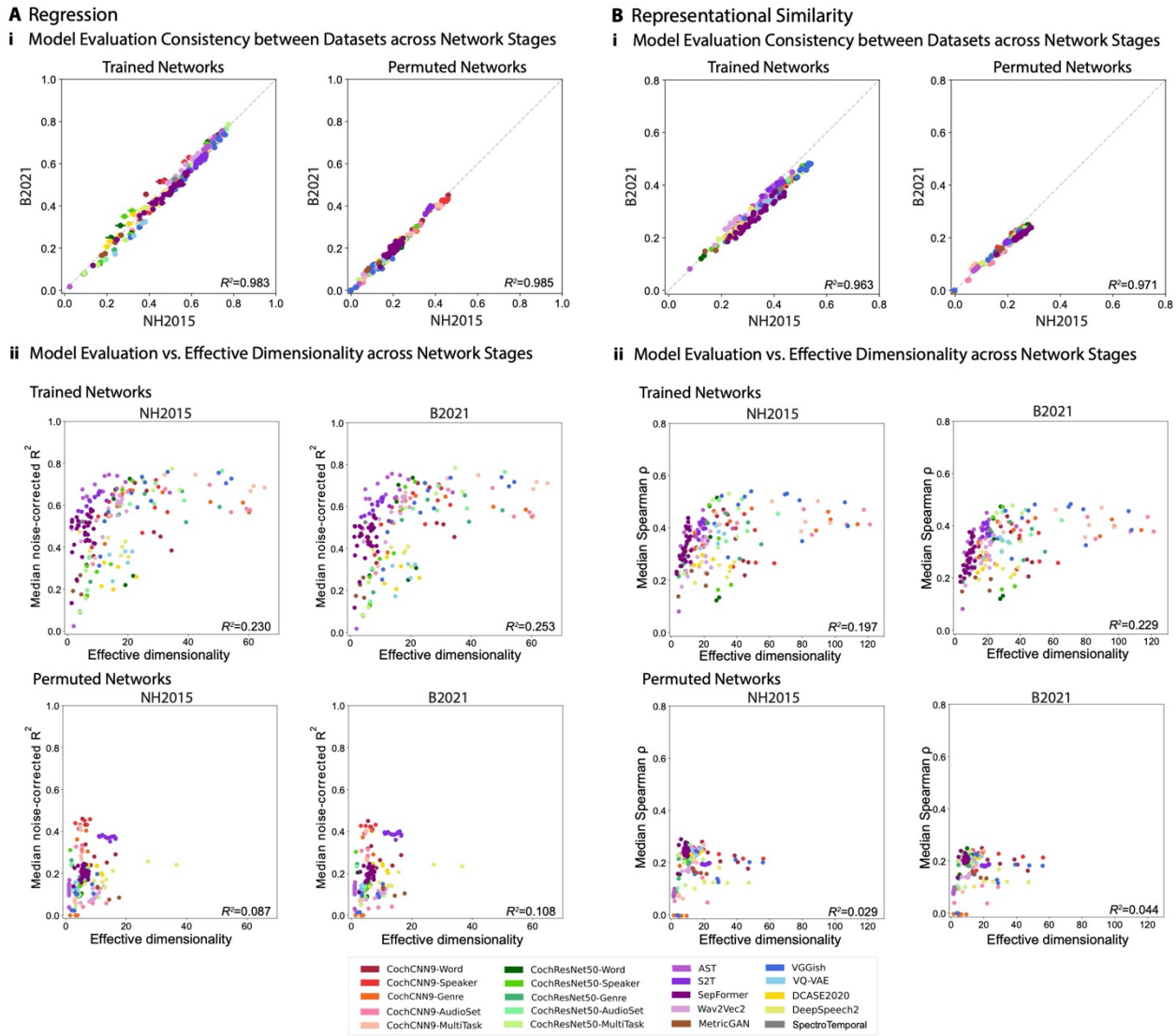


Figure S6. Effective dimensionality (ED) in relation to brain-model similarity metrics. (A) Quantification of ED in relation to the regression-based brain-model similarity metric (voxelwise modeling). Panel **i** shows the consistency of the model evaluation metric (median noise-corrected R^2) between the two datasets analyzed in the paper (NH2015 and B2021). The consistency between datasets provides a ceiling for the strength of the relationship shown in panel **ii.** Panel **ii** shows the relationship between the model evaluation metric (median noise-corrected R^2) and ED. ED was computed as described in Methods; Effective dimensionality. Each data point corresponds to a model stage, with the same color correspondence as in Figure 2 in the main text. **(B)** Same analysis as (A) but with the representational similarity analysis evaluation metric (median Spearman correlation between the fMRI and model representational dissimilarity matrices).

Supplementary Figure S7

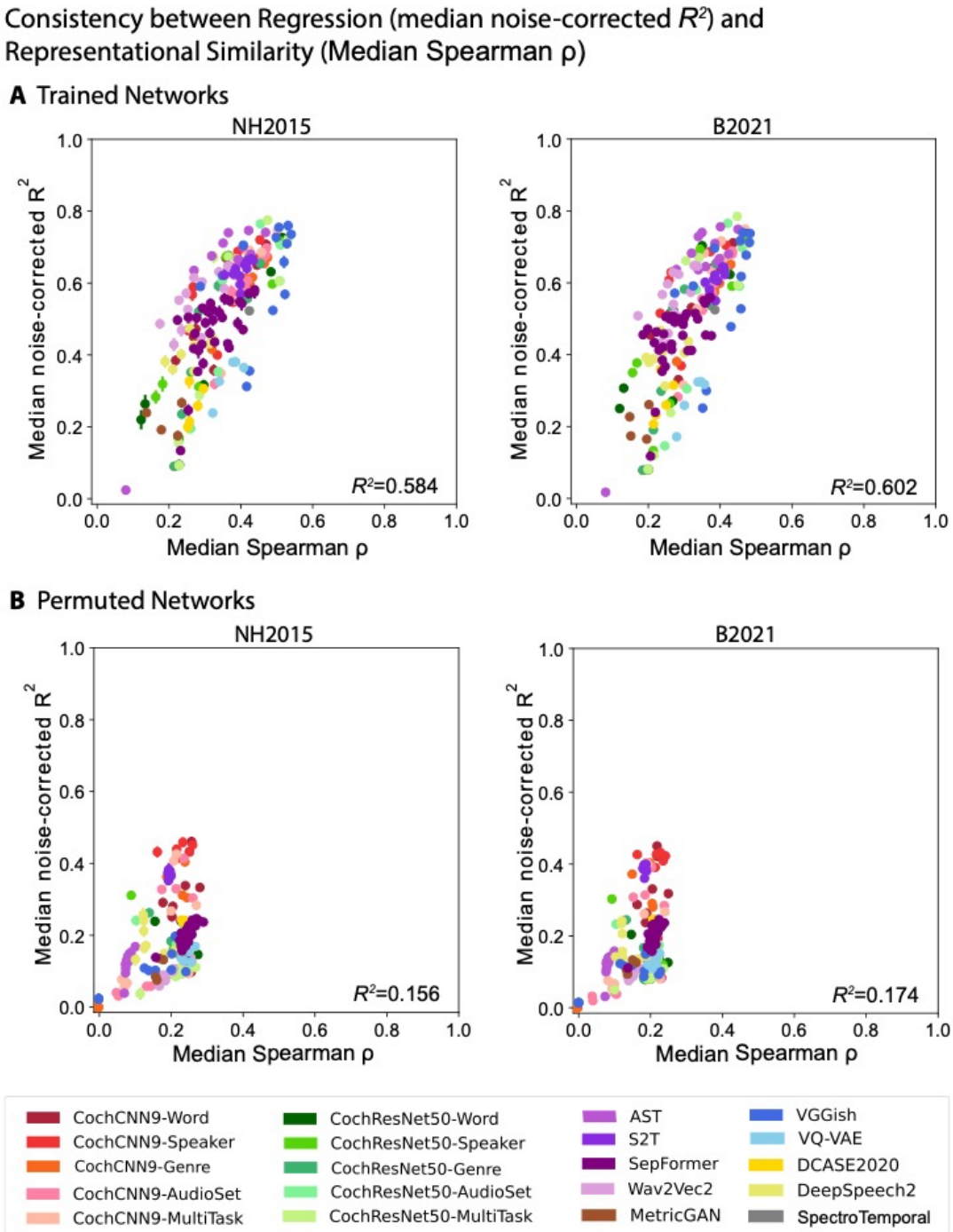


Figure S7. Consistency between regression and representational similarity brain-model similarity metrics. (A) Correlation between the regression-based metric (median noise-corrected noise-corrected R^2) and the representational similarity metric (median Spearman correlation) across trained network stages for the NH2015 and B2021 datasets. Each data point corresponds to a network stage, with the same color correspondence as in Figure 2 in the main text. **(B)** Same as in (A), but for permuted network stages.

Supplementary Table S1

Mechanical sound

alarm clock
bike bell
blender
camera snapping photos
car accelerating
car alarm
car engine starting
car horn
cash register
phone vibrating
clock ticking
coin in vending machine
cutting with scissors
dial tone
telephone dialing
doorbell
electric hand drill
hair dryer
helicopter
microwave running
motorcycle revving
airplane taking off
printing
radio or tv static
school bell
electric shaver
siren
telephone ringing
train warning bell
train whistle
truck beeping while backing up
typing
vacuum
car power windows
zipper
sports arena buzzer
busy signal
computer startup sound
ringtone

Nature sound

wind
water splashing
thunder
stream

Environmental sound

basketball dribbling
boiling water
car skidding
chair rolling
chimes in the wind
chopping food
coin dropping
crumpling paper
dishes clanking
water dripping
flag flapping
flushing
frying
hammering
road traffic
kettle whistling
keys jingling
newspaper rustling
pouring liquid
pouring water out of bottle
whistle
shuffling cards
spraying
tearing
squeaky toy
velcro
running water

Animal non-vocalization

bees buzzing
cicadas
crickets
dog drinking
wings flapping

Animal vocalization

cat meowing
cat purring
dog barking
puppy whining
duck quack
frog croaking
geese
crow
songbird
dog panting

Human, non-vocal sound

applause
biting & chewing
finger tapping
door knocking
walking on leaves
running up stairs
scratching
swimming
toothbrushing
walking on gravel
walking on hard surface
walking with heels
writing on paper
rubbing hands
heart beat

Human non-speech vocalization

person screaming
baby crying
breathing
coughing
crowd cheering
baby crying
gargling
grunting and groaning
humming
laughing
whistling
baby babbling
crowd laughing

English speech

background speech
boy speaking
girl speaking
man speaking
baby talk
angry shouting
whispering
woman speaking
sports announcer
computer-synthesized speech

Instrumental music

electric bass
big band music
bluegrass
blues band
cello
church bells
drum roll
drum solo
guitar
orchestra music
piano
rock guitar solo
saxophone jazz solo
horror film sound effects
cymbal crash
techno
trumpet jazz solo
video game music
violin
latin music
movie sad soundtrack
western soundtrack
action scene soundtrack
cartoon sound effects

Song (instrumental with vocals)

country song
heavy meta song
contemporary r&b song
rap song
contemporary rock song
classic rock
contemporary pop song
song from musical
punk song
reggae song
soul song

Foreign speech

spanish
french
italian
german
chinese
hindi
russian

Table S1. Natural sound stimulus set. List of all 165 sounds presented to human listeners while in the fMRI machine. Category assignments were based on judgments of human subjects on Amazon Mechanical Turk[1].
Table re-printed from Kell et al., (2018)[2].

Supplementary References

1. Norman-Haignere, S. V., Kanwisher, N. G. & McDermott, J. H. Distinct cortical pathways for music and speech revealed by hypothesis-free voxel decomposition. *Neuron* **88**, 1281–1296 (2015).
2. Kell, A. J. E., Yamins, D. L. K., Shook, E. N., Norman-Haignere, S. V. & McDermott, J. H. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron* **98**, 630-644.e16 (2018).