UNIVERSITY OF CALIFORNIA SAN DIEGO

From Production to Perception: Computational and Behavioral Characterization of Songbird Vocalizations

A Dissertation submitted in partial satisfaction of the requirements
for the degree Doctor of Philosophy

in

Bioengineering

by

Shukai Chen

Committee in charge:

Professor Gert Cauwenberghs, Chair
Professor Tim Gentner, Co-Chair
Professor Henry Abarbanel
Professor Vikash Gilja
Professor Gabriel Silva

2022

The Dissertation of Shukai Chen is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

I'd like to dedicate this dissertation to my parents.

They gave me the name Shukai, meaning comfort and happiness in Chinese, because all they ever wanted was for me to lead a healthy, comfortable, and most importantly happy life. They did their best at setting me up for success by providing me with the best educational resources any child could ask for, but at the end of the day still only wanted me to lead a joyful life and do what I love every day.

Words can't express my gratitude, but I truly believe I could not have come this far without them.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

Thanks to my parents, for supporting me throughout my years of study, and for teaching me to always try my best.

Thanks to David, for always cheering me up and telling me I'm doing great whenever I felt defeated in my projects.

Thanks to my PI, Tim, for years of mentorship that I deem one of the most precious memories throughout my PhD. I vividly remember when I first met with Tim to talk about my research interests and potentially joining for a rotation, instead of telling me what projects would work best for the lab, he asked me what he could do to help me achieve my own research interests. Similar conversations happened all the time whenever I hit a wall in my projects and went to Tim for advice—it made me realize that a good mentor like Tim centers around the student by setting them up for success rather than the other way around. I strive to be a good mentor like Tim if I have the honor of mentoring someone in the future.

Thanks to my lab. Zeke, for being the best postdoc I've ever met, and to quote one of our former lab mates, "every PhD needs a Zeke". Srihita, for being a strong and supportive partner in crime, quite literally because sadly we were committing bird murders even though it was in the name of science. Trevor, for helping me with various problems I encountered during chronic experiments. Michael, Anna, Brad and Tim, for guiding me through behavior experiments and helping me troubleshoot the apparatus. Marvin, Daril, Pablo, Katie, Lauren, Colin, and Emily, for being a constant source of encouragement and advice.

Thanks to my cohort friends, especially Xin, Josh, David, Billy, Bin, Anika, Qiuyang, Richard, Lily, Lynn, for making my life at UCSD a wonderful memory to look back on. I could've never done this without you all.

Thanks to my friends from various stages of my life, Sean, Charles, Shen, Evan, Shan, Sara, Elaine, and Scarlett, for telling me I'm funny when I tell terrible jokes.

Finally, thanks to my co-PI, Gert, for always making sure I'm on track throughout my PhD, as well as the rest of my committee, Vikash, Gabe and Henry, for their constant guidance and insightful advice.

Chapter 1, in part, is a reprint of the material as it appears in Current Biology 2021. Arneodo, Ezequiel M.; Chen, Shukai; Brown, Daril E. II; Gilja, Vikash; Gentner, Timothy Q. This dissertation author was the secondary investigator and co-author of this paper.

Chapter 2, in full, in part, has been submitted for publication of the material as it may appear in PNAS 2022, Chen, Shukai; Thielk, Marvin; Gentner, Timothy Q., 2022. The dissertation author was the primary researcher and author of this paper.

Chapter 3, in part, is currently being prepared for submission for publication of the material. Chen, Shukai; Gentner, Timothy Q. The dissertation author was the primary researcher and author of this material.

VITA

2015 Bachelor of Science in Bioengineering, Rice University

2022 Doctor of Philosophy in Bioengineering, University of California San Diego

ABSTRACT OF THE DISSERTATION

From Production to Perception: Computational and Behavioral Characterization of Songbird Vocalizations

by

Shukai Chen

Doctor of Philosophy in Bioengineering

University of California San Diego, 2022

Professor Gert Cauwenberghs, Chair
Professor Tim Gentner, Co-Chair

Songbird vocalizations are complex in nature and rich in information. Parametrizing such high-dimensional signals and extracting embedded information is an important yet difficult task. We approach this problem from three unique angles, incorporating modern state-of-the-art computational tools such as machine learning. We first explore the possibility of characterizing birdsongs with neural activities during song production. We use a recurrent neural network to parametrize zebra finch songs from past spiking activities in HVC. We show that the high-

quality song reconstruction is a direct result of the recurrent neural network. While the neural network excelled at learning high-dimensional data, we realize that the distance function commonly used on birdsongs is neither perceptually accurate nor robust to local perturbation. As a solution, we propose the auditory perceptual distance, a computational distance function that characterizes animal vocalizations with acoustic features learned by a convolutional neural network. By training the network on data collected from behaving European starling, we argue the distance function is not only perceptually accurate and robust to local noises, but also highly tunable to a user's data. Lastly, we seek to better understand the acoustic features used by songbirds, specifically European starlings, to achieve singer recognition. Through both supervised and unsupervised machine learning techniques, we prove vocal textures, characterized by summary statistics, carry a significant amount of singer information and can potentially be used as a vocal signature. By probing trained starlings with familiar textures in behavior experiments, we verify their capability of recognizing familiar singers through their vocal textures. In conclusion, this thesis explores different various ways of characterizing songbird vocalizations to extend our understanding of birdsong production and perception. The pipelines used can also be easily transferred to other species' vocalizations and has many practical applications.

CHAPTER 1

Abstract

State-of-the-art brain machine interfaces (BMIs) succeed at decoding behavioral outputs from brain activity by mapping neuronal activity onto a motor space. While abundant studies focus on limb-based actions, little has been achieved with speech protheses. In this study, we propose a songbird speech prothesis that exploits the similarities between human speech and birdsong, as well as the recent advancements in machine learning techniques. We show that a long short-term memory (LSTM) network can be trained to establish a mapping between HVC signals and zebra finch songs. By comparing our results to reconstructions from a similarly structured feed forward neural network (FFNN), we argue that the LSTM is essential to high quality reconstructions. Lastly, we design a shuffling task to verify the validity of our approach since disrupting the target signal results in noisy reconstructions.

Introduction

State-of-the-art brain machine interfaces (BMIs) succeed at decoding behavioral outputs from brain activity by mapping neuronal activity onto a motor space (Gilja et al., 2012; Gilja et al., 2015). While abundant studies focus on limb-based actions such as moving a cursor on a screen, little has been achieved with speech protheses. Preliminary studies where patients were implanted with electrodes for clinical assessments demonstrated the potential to decode speech; however, such studies on human are not only costly, but more importantly limited due to technological and logistical restrictions on human experiments (Leuthardt et al., 2011; Chartier et al., 2018; Anumanchipalli et al., 2019).

Meanwhile, songbirds have been an important model for learned complex vocal behavior due to many similarities shared between birdsong and human speech. Namely, one crucial similarity that we propose to exploit is that like human speech, birdsong is temporally structured. Specifically, we choose to study zebra finch because of the highly stereotyped nature of their motifs, which refer to a frequently repeated sequence of 3-10 syllables. Not only is the stereotyped motif structure ideal for proof of concept, studies show that repeated motifs align with bursts of HVC neurons that occur with diverse degrees of sparseness and precision (Picardo et al. 2016).

In parallel, machine learning researchers design recurrent neural networks (RNNs) to tackle tasks involving time sequence data, such as language translation and speech recognition (Wu et al., 2016, Sak et al., 2014). While traditional deep neural networks assume that inputs and outputs are independent of each other, the output of recurrent neural networks depend on the prior elements within the sequence. However, this sequential dependence also results in the infamous vanishing gradient problem where the gradients used to update the weights shrink exponentially. Long short-term memory (LSTM) networks are proposed as a special type of RNN that partially solve the vanishing gradient problem, because LSTM units allow gradients to also flow unchanged (Hochreiter & Schmidhuber, 1997). In this study, we propose a songbird speech prothesis that uses LSTM to map neural signals to birdsongs.

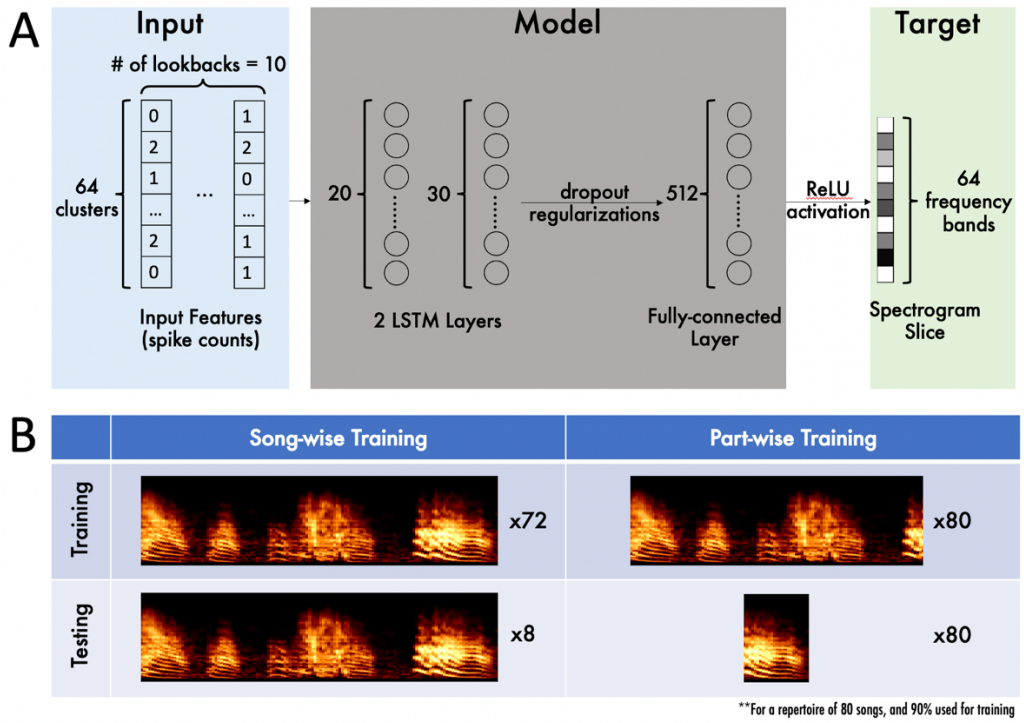

Figure 1.1 Model architecture and training schemes.

A) LSTM architecture. See Materials and Methods for detailed descriptions.
B) Two different training schemes. (Left) song-wise training means keeping each motif intact while dividing the total number of motifs into training and testing sets. (Right) part-wise training means keeping the total number of motifs fixed and segmenting each motif into training and testing parts.

Results

To obtain HVC activity and vocal output, we implant 16- or 32-channel Si-probes in male, adult zebra finches and simultaneously record extracellular voltages and the birds singing.

We train a LSTM to translate neural activity directly onto song. The goal of the network is to predict the values of the spectral components of the recorded song at a time bin t_i, given the values of neural activity features over a lookback window of M previous time bins ($t_i, t_{i-1}, \ldots t_{i-M+1}$) (Fig. 1.1A). The neural activity is fed in the form of an array of mean firing rates over each time bin, of each putative unit/multiunit sorted from the recordings (64 clusters in total). The spectral components of the song are represented by the energy across 64 mel frequency bands (Fig. 1.1A). For each session (day) of recording, we sort the spikes and separate the renditions of a song motif and the corresponding neural activities into non-overlapping training, validation, and test datasets. We then train our network and decode the spectral components from a test set and generate synthetic motifs of song.

We segment our dataset in two ways and train a LSTM on training and testing datasets yielded from each segmentation scheme (Fig. 1.1B). First, for "song-wise" training, we keep each motif intact and divide the number of motifs into, for example, 90% for training and 10% for testing. Second, for "part-wise" training, we segment each motif into, for instance, 30 parts of equal lengths and use the same part from all motif renditions for testing, and the other 29 parts for training. The goal of part-wise training is to examine the model's ability to generalize to new parts of the motif that it has not been exposed to.

In addition, by training a separate feed-forward neural network (FFNN) of a similar size as the LSTM, we establish a performance baseline for computationally inexpensive training as well as investigate the necessity of the LSTM architecture.

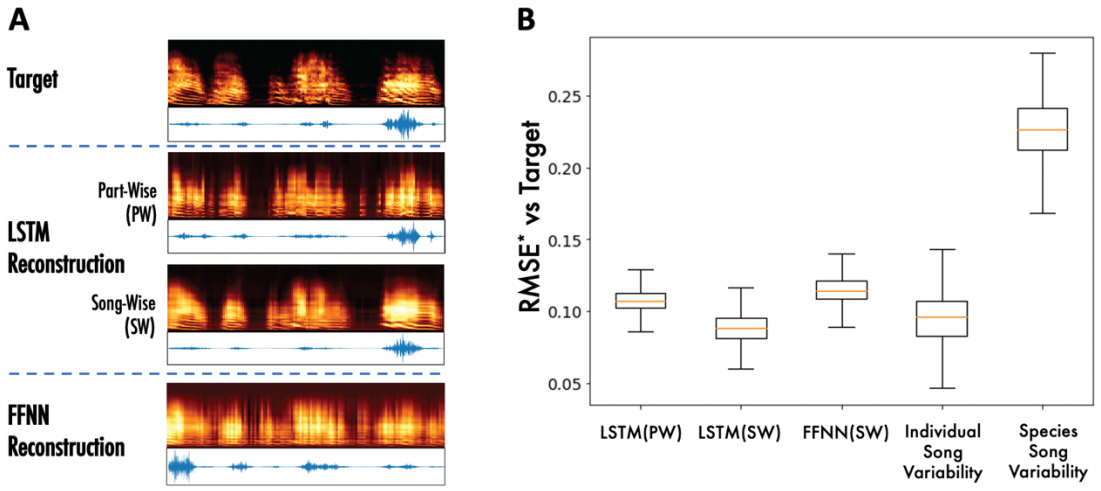

Figure 1.2 Comparing model reconstructions under different architectures and training schemes

A) From top to bottom: example target song spectrogram and its corresponding sound pressure waveform; LSTM-reconstructed spectrogram under part-wise training and its corresponding sound pressure waveform; LSTM-reconstructed spectrogram under song-wise training and its corresponding sound pressure waveform; FFNN-reconstructed spectrogram under song-wise training and its corresponding sound pressure waveform.

B) LSTM(PW): RMSE between each pair of part-wise-trained LSTM reconstruction and its target spectrogram. LSTM(SW): RMSE between each pair of song-wise-trained LSTM reconstruction and its target spectrogram. FFNN(SW): RMSE between each pair of song-wise-trained FFNN reconstruction and its target spectrogram. Individual song variability: RMSE between each pair of natural songs from the singer's repertoire. Species song variability: RMSE between each pair of natural songs from the species' song dataset.

To evaluate the performance of the trained models, we generate synthetic motifs and compare them to target natural motifs by calculating the root mean square error (RMSE). We also compute the individual song variability as pairwise RMSEs among all target natural motifs. Another metric we compare our results with is the species song variability, defined as the pairwise RMSEs among motifs from multiple zebra finch singers.

The conclusions from our results are three-fold. The first key finding from our analysis is that a mapping can be established computationally between HVC activities and zebra finch songs regardless of training method or network architecture. All reconstructions, including LSTM under song-wise training (mean=0.088, std=0.011), LSTM under part-wise training (mean=0.108, std=0.008), and FFNN under song-wise training (mean=0.115, std=0.009), fall within the range of individual song variability (mean=0.095, std=0.018, range=[0.036, 0.151], Fig. 1.2B). Moreover, all reconstruction RMSEs are significantly lower than the measured species variability (mean=0.225, std=0.022).

Secondly, the model is able to generalize the learned mapping to novel song elements. While song-wise training achieves a lower RMSE on average than part-wise training, part-wise training still yield a relatively low RMSE, indicating its capability of generalizing to novel song elements at a high fidelity (Fig. 1.2B). In fact, such difference conforms to our intuition as it is more difficult to generalize to unfamiliar data than to predict a different rendition of the same motif.

Lastly, we show that the LSTM is essential to high quality reconstructions. Although trained under the same conditions, the FFNN and the LSTM show significant differences in reconstruction quality ($p<0.05$, t-test) where the FFNN yield higher RMSEs on average, meaning its reconstruction is noisier. The disparity is obvious at a glance of the reconstructed spectrograms: FFNN reconstructions are significantly noisier than the LSTM ones, showing signs of time-averaging and incorrect predictions on silences (Fig. 1.2A). Since HVC is commonly believed to control the timing of singing activities, the FFNN's poor performance on predicting silences suggest that it is incapable of fully capturing the information embedded within HVC activities.

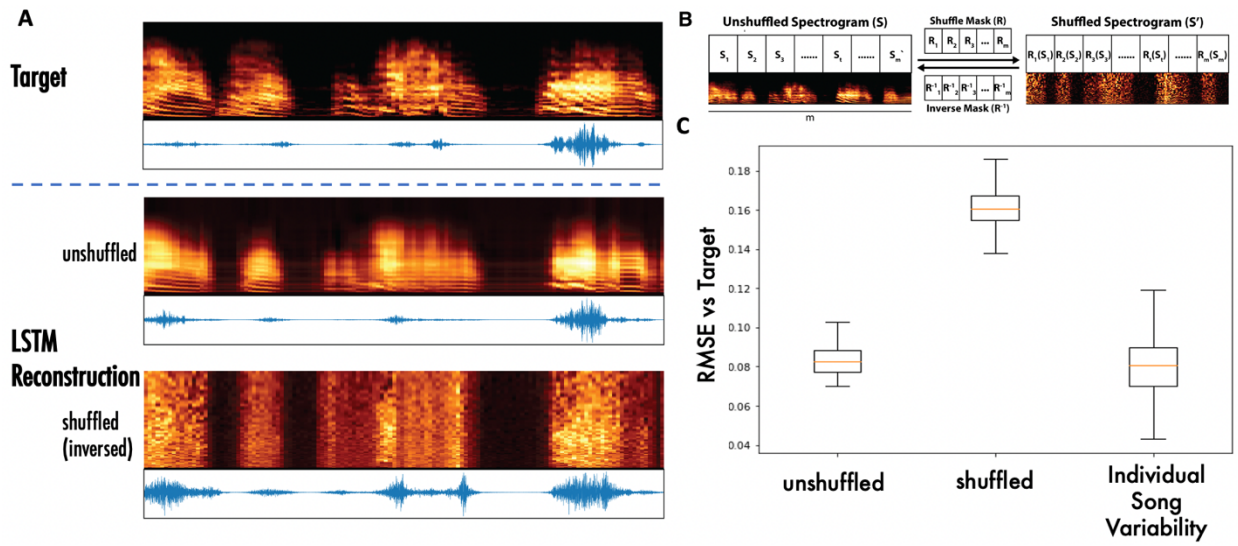

Figure 1.3 The shuffling mask disrupts the model's reconstruction

A) From top to bottom: target spectrogram and its sound pressure waveform; reconstructed spectrogram from LSTM trained on spectrograms without the shuffling mask, and its corresponding sound pressure waveform; reconstructed spectrogram from LSTM trained on spectrograms with the shuffling mask (reverted for easy visualization), and its corresponding sound pressure waveform.

B) Conversion between unshuffled and shuffled spectrograms. The same shuffling mask and inversion mask are applied to every spectrogram.

C) Unshuffled: RMSE between each pair of reconstruction and target spectrograms from LSTM trained on unshuffled spectrogram. Shuffled: RMSE between each pair of reconstruction and target spectrograms from LSTM trained on shuffled spectrogram. Individual song variability: RMSE between each pair of natural songs from the singer's repertoire.

Now that we prove it is possible to establish a mapping between HVC activities and birdsongs, we introspectively question whether such mapping is due to the LSTM exploiting the stereotypical nature of zebra finch song and outputting the same reconstruction regardless of inputs. To examine the validity of our findings, we apply a shuffling mask to permute all elements spectrally while preserving the temporal structure consistent across all renditions of the motif (Fig. 1.3B). The result is renditions of the same noise-like pattern that do not resemble birdsong spectrograms but are stereotyped. If the LSTM were only outputting the same spectrogram regardless of the input, it should be able to output the noise like pattern as well. After training the LSTM on shuffled spectrograms, we revert its predicted spectrograms by reversing the shuffling mask. The reverted reconstructions show no spectrogram like features (Fig. 1.3A) and the LSTM yields significantly worse reconstruction than the LSTM trained on natural, unshuffled birdsongs (Fig. 1.3C). These results prove the validity of our previous finding that the LSTM is capable of establish a mapping between HVC activities and birdsongs both temporally and spectrally.

Conclusion

In conclusion, we have demonstrated a speech prothesis for birdsong, using state-of-the-art machine learning techniques. We show that our LSTM-based model is able to establish a mapping between HVC activities and birdsongs both temporally and spectrally. In addition, we prove that the LSTM is essential to high quality reconstructions in that a FFNN of similar structure fails at fully capturing the information embedded within HVC activities. This study provides a proving ground for vocal prosthetic strategies in human because of the many similarities between human speech and birdsongs.

Materials and Methods

Neural network training

Neural network-based decoders were coded in python, using Tensorflow. They were run on PCs equipped with NVidia GPUs (Tesla k40, Titan Z, and Titan X Pascal).

LSTM network architecture

The network has 2 layers of LSTM cells, with 30 units in the first layer and 20 in the second. The output layer has 64 units, each for a mel spectrogram band. Both LSTM layers utilized 20% dropout and 0.001 L2 regularization during training to prevent overfitting.

Feed-forward Network architecture

The architecture is essentially the same as that of the LSTM network, but it replaces the LSTM layers with one dense layer of RELU units (Rosenblatt, 1958), which halves the dimension of the input vector. The hidden layer utilized 20% dropout and 0.001 L2 regularization during training to prevent overfitting.

Training procedure

We utilize a gradient-based optimizer, Adam/rmsprop (Kingma & Ba, 2015), and mean square error (MSE) as a loss function for LSTM/FFNN. Two training conditions are experimented, referred to as song-wise and part-wise training.

Song-wise training: we use 10% of all the motifs for testing and the rest motifs for training. We make 10 passes using non-overlapping motifs as testing set, in order to have as many decoded examples as number of motifs in the session. In each pass, all the neural-activity/decoder-target pairs (one per bin) are fed in random order to the network, both during training and decoding.

Part-wise training: we subsequently leave out a fraction of each motif when training (roughly 3.3%), train on the complement and generated the song corresponding to the masked fraction. We repeat this procedure tiling the whole motif and generate entire motifs using segments of data that were novel to the decoder.

In both training conditions, 10% of the training set was reserved as validation set for early stopping, where the training session would be stopped if validation loss failed to decrease within 5/10 training epochs.

Spectrogram inversion

We used LSEE-STFTM algorithm to invert spectrograms back to audio waves (Griffin & Lim, 1984).

Spectrum shuffle mask

Time warping

We adopted a simplified version of Dynamic Time Warping (DTW, Anderson et al., 2018) specific to zebra finch songs. Instead of segmenting the song into different syllables and matching each syllable to different syllable templates, we took advantage of the stereotypical nature of zebra finch songs and directly computed minimal distance matrices (D) between each song-level spectrogram and a spectrogram template. Starting at the first slice of each spectrogram,

$$D(i,j) = d(i,j) + minD(i-1,j) \ \ iff \ wj(l-1) \neq wj(l-2), D(i-1,j-1), D(i-1,j-2)$$

Where i indexes the time frames of the input pattern, j indexes the time frames of a single template, l indexes the ordered steps along a specific path. d(i, j) is the local distance between slice i and slice j. $w_j(l)$ denotes the specific step at l in the space of j. Once a distance matrix D was calculated, we determined an optimal path with the lowest cumulative distance between the input and the template, and proceeded to stretch, delete or keep each input slice, depending on the path.

Masking

We applied a random yet consistent shuffling mask, P, to our entire warped spectrogram repertoire so that spectral consistency across time is disrupted while the temporal pattern within each motif remains. For the i-th spectrogram slice in each warped song, we shuffled all 64 spectral elements using the same shuffling pattern, Pi. Treating all spectrograms with the same shuffling mask Penabled us to determine whether our model is decoding the spectral information within birdsongs, or recreating the same pattern

regardless of spectral consistency across time. In our shuffling training session, we used the shuffled spectrograms as output.

Reverting mask

After training, we tested our model on novel neural data, the target of which were also shuffled spectrograms. In order to visually compare our model's performance with and without shuffling, we reordered the reconstructed shuffled spectrograms. We achieved this by applying a reordering mask, R, that traces and reverses all the shuffling done through the aforementioned shuffling mask P. For any spectrogram S, R(P(S)) = S.

Performance Evaluation

RMSE

We used RMSE between each pair of original and predicted spectrogram magnitude as a metric to evaluate the performance of our models.

Spectral correlation

To obtain the spectral correlation across time for a pair of spectrograms, we computed the pearson correlation coefficient between each corresponding pair of spectral slices that conform the two spectrograms (via the function pearsonr from the stats module of the scipy python package Virtanen et al., 2020).

Spectrogram Normalization

In order to account for variations among motifs from different birds, we normalized spectrograms for each bird so that the collection of original spectrograms for each bird had a maximum power of 1 and minimum power of 0:

$$pi = \frac{pi - pmax}{pmax - pmin}$$

Where pi is the power of a point on either an original spectrogram or a predicted spectrogram before normalization, while pi is the normalized power of the corresponding point. pmax denotes the maximum power of the entire set of original spectrograms, while pmin represents the minimum power of the entire set of original spectrograms. With such normalization, we were able to account for variations among motifs from different birds while keeping the variations within motifs from the same bird.

Acknowledgement

Chapter 1, in part, is a reprint of the material as it appears in Current Biology 2021. Arneodo, Ezequiel M.; Chen, Shukai; Brown, Daril E. II; Gilja, Vikash; Gentner, Timothy Q. This dissertation author was the secondary investigator and co-author of this paper.

References

Anderson P, Wu Q, Teney D, Bruce J, Johnson M, Sünderhauf N, et al. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments [Internet]. arXiv.org. 2018 [cited 2022Aug8]. Available from: https://arxiv.org/abs/1711.07280

Anumanchipalli GK, Chartier J, Chang EF. Speech synthesis from neural decoding of spoken sentences. Nature. 2019;568(7753):493–8.

Chartier J, Anumanchipalli GK, Johnson K, Chang EF. Encoding of articulatory kinematic trajectories in human speech sensorimotor cortex. Neuron. 2018;98(5).

Gilja V, Nuyujukian P, Chestek CA, Cunningham JP, Yu BM, Fan JM, et al. A high-performance neural prosthesis enabled by control algorithm design. Nature Neuroscience. 2012;15(12):1752–7.

Gilja V, Pandarinath C, Blabe CH, Nuyujukian P, Simeral JD, Sarma AA, et al. Clinical translation of a high-performance neural prosthesis. Nature Medicine. 2015;21(10):1142–5.

Griffin D, Lim J. Signal Estimation from modified short-time Fourier transform. ICASSP '83 IEEE International Conference on Acoustics, Speech, and Signal Processing. 1984;

Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation. 1997;9(8):1735–80.

Kingma D, Ba J. Adam: A method for stochastic optimization [Internet]. arXiv.org. 2015 [cited 2022Aug8]. Available from: https://arxiv.org/abs/1412.6980v8

Leonardo A. Ensemble coding of vocal control in birdsong. Journal of Neuroscience. 2005;25(3):652–61.

Leuthardt EC, Gaona C, Sharma M, Szrama N, Roland J, Freudenberg Z, et al. Using the electrocorticographic speech network to control a brain–computer interface in humans. Journal of Neural Engineering. 2011;8(3):036004.

Picardo MA, Merel J, Katlowitz KA, Vallentin D, Okobi DE, Benezra SE, et al. Population-level representation of a temporal sequence underlying song production in the Zebra Finch. Neuron. 2016;90(4):866–76.

Rosenblatt F. The Perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review. 1958;65(6):386–408.

Sak H, Senior A, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Interspeech 2014. 2014;

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental algorithms for scientific computing in python [Internet]. Nature News. Nature Publishing Group; 2020 [cited 2022Aug8]. Available from: https://www.nature.com/articles/s41592-019-0686-2

Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, et al. Google's Neural Machine Translation System: Bridging the gap between human and machine translation [Internet]. arXiv.org. 2016 [cited 2022Aug8]. Available from: https://arxiv.org/abs/1609.08144

CHAPTER 2

Abstract

Studies comparing acoustic signals often rely on pixel-wise differences between spectrograms, as in for example mean squared error (MSE). Changes in pixel-wise error are not representative of perceptual sensitivity, however, and these functions can be highly sensitive to small local signal changes that may be imperceptible. In computer vision, high-level visual features extracted with convolutional neural networks (CNN) can be used to calculate the fidelity of computer-generated images. Here, we propose the auditory perceptual distance (APD) model based on acoustic features extracted with an unsupervised CNN and validated by perceptual behavior. Using complex vocal signals from songbirds, we trained a Siamese CNN on a self-supervised task using spectrograms rescaled to match the peripheral frequency sensitivity of European starlings, Sturnus vulgaris. We define APD for any pair of sounds as the cosine distance between their corresponding feature vectors extracted by the trained CNN. We show that APD is more robust to temporal and spectral translation than MSE, and captures the sigmoidal shape of typical behavioral psychometric functions over complex acoustic spaces. When fine-tuned using starlings' behavioral judgments of naturalistic song syllables, the APD model yields even more accurate predictions of perceptual sensitivity, discrimination, and categorization on novel complex (high-dimensional) acoustic dimensions, including diverging decisions for identical stimuli following different training conditions. Thus, the APD model outperforms MSE in robustness and perceptual accuracy, and offers tunability to match experience dependent perceptual biases.

Introduction

Characterizing and comparing natural acoustic signals in a manner that mirrors perception is vital for researchers across wide-ranging fields spanning neuroscience, artificial intelligence, and psychology. These sounds, which include vocal and other acoustic communication signals as well as environmental sounds, typically vary simultaneously along multiple physical dimensions each of which may carry different (or no) behaviorally relevant information, which in turn may vary across contexts. Species differences complicate matters even further, as the features that carry perceptual relevance for one species may not necessarily generalize to another (Dooling & Prior, 2017). Thus, an ideal measure of the perceptually relevant similarities and differences between natural acoustic signals should be able to capture the complex multi-variate features spaces of these sounds, and have a flexibility that permits tuning to species- and context-specific functional outputs.

To quantify differences between two audio signals, current studies, especially those using machine learning for audio generation, often resort to pixel-wise error functions between corresponding spectrograms, such as mean squared error (MSE) (Arneodo et al., 2021; Akbari et al., 2018; Purwins et al., 2019). This approach has the benefit of easy quantification, but deviates from perception in two significant ways. First, commonly used spectrograms in the linear frequency (Hz) scale or the mel scale are not representative of perceptual frequency sensitivity (Stevens et al., 1937). In many animals, including both humans and European starlings (Fig. 2.1A), frequency sensitivity, defined as the minimum detectable change in frequency, is not uniform across the frequency spectrum but rather scales positively with frequency (Kuhn et al., 1980). That is, a 20 Hz frequency deviation at 8,000Hz is perceptually less noticeable than a change of 20Hz at 1,000 Hz, although the absolute values of frequency deviations are the same.

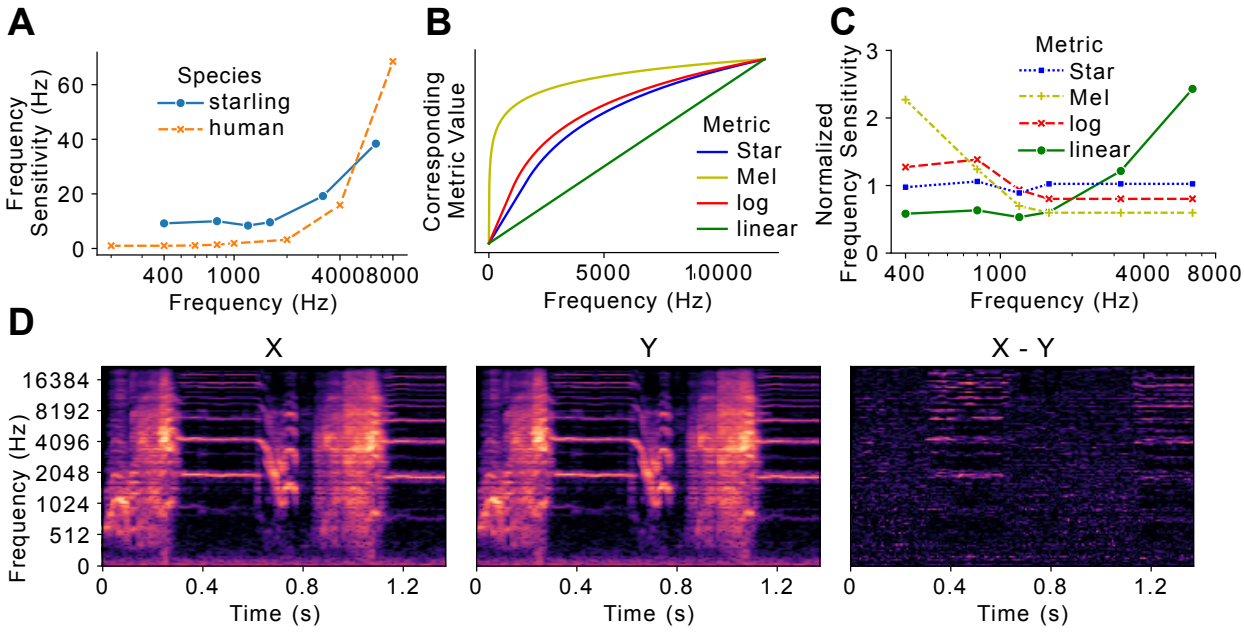

Figure 2.1 Flaws of the current approach

A. Frequency sensitivity in both starlings and humans. Frequency sensitivity is defined as the minimum frequency change to be discernible by an animal.
B. Mapping between different frequency metrics and the Hz scale.
C. Normalized frequency sensitivities in each frequency scale. Sensitivity values are normalized to the mean within each frequency scale.
D. Two minimally offset spectrograms and their pixel-wise distance. Two Star-scale spectrograms (X and Y) are the same but offset from each other by only one frequency band, corresponding to one Star. The per-pixel distance spectrogram (X-Y) is calculated with pixel-wise subtraction between X and Y.

Mel-scale spectrograms, while mitigating the discrepancy, still fall short at compensating for the non-uniformity of animal frequency sensitivity. At the same time, MSE and pixel-wise errors in general do not capture the perceptual differences between two signals, but rather focus on local details. The misrepresentation caused by MSE is evident in Fig. 2.1D, where two spectrograms offset from each other by only 5ms of silence, resulting in significant per-pixel errors across all frequency bands, whereas the perceptual error conveyed by the animal's auditory cortex should be minimal. We are in desperate need of an auditory distance metric that is representative of an animal's perception.

Recent studies in computer vision have shown success at quantifying the visual distance between two images based on high-level features (Mahendran & Vedaldi, 2015; Dosovitskiy & Brox, 2016; Gatys et al., 2016; Johnson et al., 2016). Instead of calculating per-pixel differences between two images, these error functions extract and compare embeddings with the help of convolutional neural networks (CNN). These embeddings are successful at capturing significant global features, which can be utilized as quantifiable measures of perceptual distances between images. By using the distance between feature vectors as a loss function, deep neural networks, particularly ones aiming at image generation, have achieved feature visualization (Mahendran & Vedaldi, 2015), texture synthesis (Gatys et al., 2016), and image style transfer (Johnson et al., 2016). However, these CNNs' success at learning visual features relies heavily on the availability of massive labeled image datasets such as ImageNet (Deng et al., 2009). Such knowledge is not directly transferable as it simply cannot be presumed that spectrograms and visual images share the same feature space. In fact, it has been shown that the learned features from ImageNet do not transfer well to fine-grained tasks (Kornblith et al., 2019). To apply

transfer learning to spectrograms, we are in need of large labeled bioacoustic datasets to train the CNN on, which are scarce due to the costly nature of manual labeling.

In parallel, researchers in various fields facing the same issue find success in applying self-supervised learning to unlabeled datasets that are readily available (Zhai et al., 2019; Noroozi & Favaro, 2016; Jing & Tian, 2021; Kolesnikov et al., 2019). Instead of using true labels as training targets, self-supervised training involves pretext tasks, meaning it applies certain automatic preprocessing to the unlabeled dataset and optimizes on corresponding machine-generated pseudo-labels. A popular self-supervised model for visual feature extraction is Jigsaw, where an image is decomposed into small randomly-ordered puzzle pieces; tasked to reorder these shuffled tiles, the network eventually learns to identify significant visual features from images (Noroozi & Favaro, 2016). A similar model has been proposed for spectrograms where the network learns spectrogram features from reordering spectrogram fragments (Carr et al., 2021). Unlike the original jigsaw model, the model on spectrograms performs the best when spectrograms are only dissected in one dimension, specifically the frequency domain.

We propose the Auditory Perceptual Distance (APD) model, a computational model to quantify perceptual distances based on high-level spectrotemporal features of acoustic signals. The APD model is designed to combine innovative machine learning approaches to tackle the shortcomings of pixel-wise error functions through a three-step process (Fig. 2.2). In this study, we focus primarily on starlings' auditory perception of vocalizations. To better portray starling perception, we devise the Star scale, a starling-specific frequency metric where one "Star" across all frequency levels represents the same perceptual distance as observed by starlings (Fig. 2.1B, Fig. 2A). Then we train a CNN to learn significant acoustic features through a self-supervised task on starling vocalization spectrograms, namely reordering shuffled frequency bands (Fig.

2.2B). All spectrograms involved are based on a redesigned frequency scale that matches the experimentally-measured perceptual sensitivity of starlings. While this process familiarizes the model with bioacoustic signals and characteristic spectrotemporal features of those signals, it is purely computational and devoid of ground truth that can only be characterized through animal experiments. Therefore, our network is fine-tuned on behavioral data from real-world experiments that only target specific aspects of auditory perception since we believe it is unreasonable to attempt to characterize the entirety of animal perception within a low-dimensional vector (Fig. 2.2C). Finally, the APD model calculates the cosine distance between acoustic feature vectors extracted by the trained CNN, thereby characterizing distances in the aforementioned perceptual space. We evaluate the absolute performance of the APD model on behavioral data where ground truths are available, as well as its relative efficacy when compared to MSE.

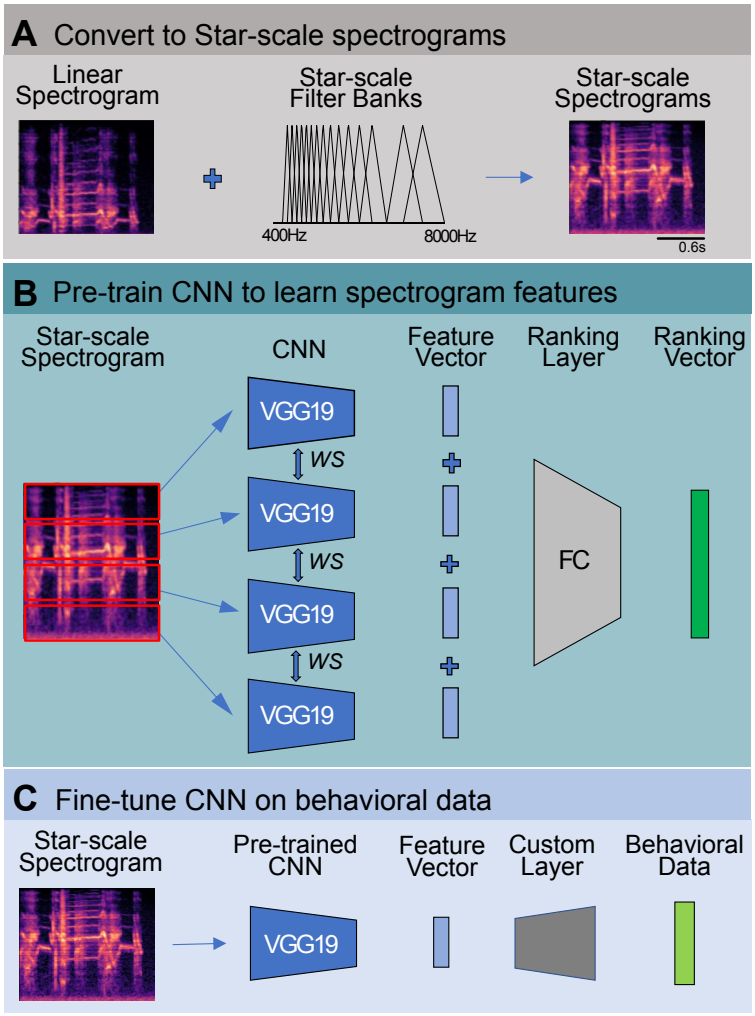

Figure 2.2 Training the APD model is a three-step process.

A. Converting to Star-scale spectrograms. All linear spectrograms are first converted to Star-scale spectrograms through a set of Star-scale filter banks. Here we show an example set of 16 Star filters, covering 400Hz to 8,000Hz; for parameters used in this study, refer to Materials and Methods.

B. Pre-training the APD model. The model is pre-trained on starling vocalization spectrograms in order to learn spectrogram features. Each spectrogram is divided in the frequency domain into four equally sized slices which are subsequently shuffled (not shown here). The model is tasked to output a ranking vector that indexes the original position of the shuffled slice. During training, all four slices are fed into the same CNN, yielding four feature vectors, which are then concatenated and passed to the fully connected (FC) ranking layers for classification.

C. Fine-tuning the APD model. Once the model is pre-trained, we can directly use the learned CNN for fine-tuning on behavioral data. All the pre-trained weights are transferred directly to the same CNN which is now connected to task-specific custom networks. Inputs to the fine-tune model are unsegmented Star-scale spectrograms. Outputs are animal judgments collected during the behavioral experiment.

Results

As outlined in Fig. 2.2, training the APD model is a three-step process, where each step takes an innovative approach to tackle a prominent shortcoming with existing methods. Therefore, we address the effectiveness of each proposed step separately and compare its performance to existing approaches. First, we inspect the perceptual uniformity of the Star scale, our custom starling-specific frequency scale, based on which we construct all spectrograms for the APD model. Next, we evaluate the effectiveness of pre-training on spectrograms compared to borrowing ImageNet weights, as well as the robustness of a pre-trained model to small local changes. Finally, we draw comparisons between computed and experimentally recorded perceptual distances after fine-tuning the model on behavioral data where animal judgments can be considered ground truth.

Star Scale

The APD model is designed to work with spectrograms based on species-specific frequency metrics in lieu of Hz-based spectrograms, in order to avoid the existing mismatch between frequency scales and animal frequency sensitivity. A perceptually accurate frequency scale should be perceptually uniform, meaning a change of one unit should be judged by listeners to be equal in distance from one another regardless of the absolute frequency values. To meet these requirements, we devise the Star scale, a starling-specific frequency metric where one Star across all frequency levels represents the same perceptual distance observed by starlings (Fig. 2.1C). To compare the Star scale to existing frequency scales, namely the Hz scale and the mel scale, we compute frequency sensitivity measurements collected by Kuhn et al. in units of Star and mel (Fig. 2.1C; Kuhn et al., 1980). Sensitivity values are normalized to the mean within each frequency scale, as we focus more on the fluctuation of sensitivity across all frequency levels rather than the absolute values. Out of three frequency scales, the Hz scale shows the

highest degree of variability (μ=1.0, σ=0.75, range 0.53~2.43) followed by the mel scale (μ=1.0, σ=0.26, range 0.80~1.38). The Star scale appears the most uniform (μ=1.0, σ=0.06, range 0.89~1.06); even the maximum degree of fluctuation, logged at 120 Star (1,200Hz), only measures 11%. These results confirm that the Star scale outperforms both the Hz scale and the mel scale in terms of perceptual uniformity at all frequencies.

Pre-training

In this experiment, we evaluate the effectiveness of the APD model trained solely on Star-scale spectrograms without tuning on animal judgments. To distinguish this model from the fully trained APD model in the next experiment, we refer to this model with only pre-training as the naive APD model and the fine-tuned model as the tuned APD model. Instead of adopting ImageNet weights like most computer vision models, we opt to pre-train our model on Star-scale spectrograms, as images and spectrograms do not necessarily share the same feature space. The pre-training dataset consists of 21,000 1.4s-long unlabeled starling vocalizations, converted to Star-scale spectrograms (Fig. 2.2B, Materials and Methods). The pretext task we choose is spectrographic Jigsaw, a spectrogram-specific adaptation of a popular self-supervised training task where networks learn high-level features through sorting the shuffled puzzle pieces of an image (Fig. 2.2B; Noroozi & Favaro, 2016; Carr et al., 2021). Once the model is trained, the dense layers are disconnected from the CNN, the output of which is a 512-dimensional feature vector. We randomly choose 30 1.4s long Star-scale spectrograms for testing.

We consider two spectrograms offset by a few rows or columns in either the frequency or the time domain, respectively. For simplicity, the offset was added as silence to one of the four edges of the spectrogram (top, bottom, left, right), and for each spectrogram, offset versions with opposite-edge offsets were compared (i.e., top vs. bottom, left vs. right). We expect the model to react differently to temporal and spectral shifts. Intuitively, two audio signals differing only in trailing or leading silence should

contain resembling information. On the other hand, while two signals offset by only a few Hz should be perceptually similar, they should sound more distinct as the frequency offset increases.

As shown in Fig. 2.3A, shifting as little as one line in the time domain results in a significant MSE (MSE=0.06 ± 0.01 normalized, N=30) when the time difference (5ms) is close to the gap detection threshold (Klump & Maier, 1989). Meanwhile, the same shift in the time domain yields an APD close to zero (APD=1.07E-4 ± 1.83E-4 normalized, N=30). In fact, the APD stays in the vicinity of zero even with longer silence padding in the time domain (APD=5.93E-4 ± 8.65E-4 normalized, N=450). A similar trend is observed in the frequency domain (Fig. 2.3B)–a slight shift of one Star, approximately the smallest frequency shift distinguishable by starlings, offsets the MSE significantly (MSE=0.06 ± 3.13E-3 normalized, N=30) (Kuhn et al., 1980). In contrast, the change in the APD is far less drastic at the beginning (APD=3.16E-3 ± 1.46E-3 normalized, N=30) while steadily increasing. Both MSE and APD measurements have been normalized so that the maximum and minimum achievable distances are 1 and 0, respectively. From these results, it is clear that the APD better conforms to our intuition and reflects starlings' perception more accurately than the MSE.

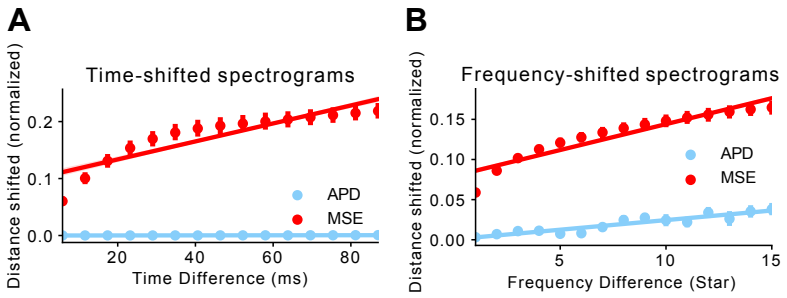

Figure 2.3 Perceptual loss is more robust to small changes than pixel-wise loss.

We measure the distance between two time/frequency-shifted spectrograms using MSE and APD and compare the trajectories of both distance metric as shifting becomes more significant.

A. Distance measured between two time-shifted spectrograms. We start with a pair of spectrograms separated by 5ms (leftmost points) and progress at a step size of approximately 5ms. We then calculate MSE and APD between each pair and fit a linear regression to all the points within either distance metric.
B. Distance measured between two frequency-shifted spectrograms. Similarly, we start with a pair of spectrograms separated by 1 Star (leftmost points) and progress at a step size of 1 Star.

Fine-tuning

While the pre-training process proves fruitful at orienting the model to better extract global features embedded in spectrograms, animal judgments are the ultimate ground truth and therefore essential for modeling animal perception. In this analysis, we fine-tune the naive APD model on artificial stimuli used in a behavioral experiment. The original experiment and its findings have been detailed elsewhere (Thielk, 2019). Briefly, starlings were trained on a two alternative choice tasks on eight natural stimuli (A~H) where four of these stimuli (for example, A, B, C,and D) were associated with a left response and the other four (E, F, G, and H in the previous example) were associated with a right response. Once the subjects achieved stable recognition on these natural stimuli, the stimuli dataset was expanded to include a series of artificial stimuli ("morph") that were, in short, machine-generated linear interpolations of the familiar natural stimuli (Fig. 2.4A). The subjects were subsequently required to classify each artificial stimuli based on their prior training on natural stimuli classification. A double staircase training procedure was used to allow the birds to place their perceptual boundaries freely among all linear interpolated morph stimuli (Materials and Methods). One of the main findings was that birds trained under the same condition (for example, peck left for ABCD, right for EFGH) yielded a remarkable degree of consensus on decision boundaries across morph stimuli, suggesting a shared perceptual space. To replicate the training process computationally, we simulate a naive bird's perceptual space with our naive APD model, and a trained bird's with our fine-tuned APD model, trained on aforementioned experimental decisions made by birds. To assess the performance of the tuned APD model, we test our models on morph stimuli and draw comparisons between the APD-simulated and the experimentally measured psychometric curves.

When we examine the computed psychometric curves, the hypothesis is that a high-performing perceptual distance metric achieves a high resemblance to the ground truth, specifically in terms of the inflection point and the Hill coefficient. The inflection point marks the decision boundary within a set of stimuli whereas the Hill coefficient entails sensitivity at the inflection point, measured as the slope of the

psychometric curve (Hill, 1910; Hill, 1913; Weiss, 1997). In an example comparison between computed and measured psychometric curves (Fig. 2.4B), fine-tuned APD is capable of yielding simulations close to the ground truth whereas there exists a significant mismatch between MSE and the ground truth, especially in sensitivity. In the comparison, we also include predictions made directly with the fine-tuned APD model, which not only uses trained APD feature vector but also takes advantage of trained classification layers. A similar trend is observed across all stimuli sets for both computational distance metrics (Fig. 2.4C-D). To characterize resemblance to the ground truth, we calculate the absolute residual error between each computed Hill coefficient and the ground truth under the same training conditions (morph stimuli, cohort, etc.), and compare it to internal variability within the ground truth (GT), computed as absolute errors between all pairs of subject judgments under the same training conditions (Fig. 2.4C). The residual errors incurred by MSE are significantly different from the ground truth variability [μ=0.160, σ=0.058, compared to GT: $p<0.001$, linear mixed effects model (LMM)] while the sensitivities of fine-tuned APD curves, including tuned APD and network predictions, are much closer to the ground truth [tuned APD: μ=0.078, σ=0.065, $p>0.1$, LMM against GT; APD NN: μ=0.068, σ=0.055, $p>0.1$, LMM against GT]. The same set of residual errors is calculated for the inflection point (Fig. 2.4D). Similar results are observed: both tuned APD metrics highly resemble the ground truth [tuned APD: μ=8.76, σ=7.85, $p>0.1$, LMM against GT; APD NN: μ=8.61, σ=7.95, $p>0.1$, LMM against GT], whereas MSE shows significant deviation from the ground truth [μ=16.17, σ=13.62, $p<0.001$, LMM against GT]. These results suggest APD is indeed a high-performing perceptual distance, yielding decision boundaries and sensitivities within the variability of the ground truth. Moreover, direct comparisons between APD and MSE provide a strong demonstration that APD is much more representative of starling perceptual sensitivity around the decision boundary than MSE.

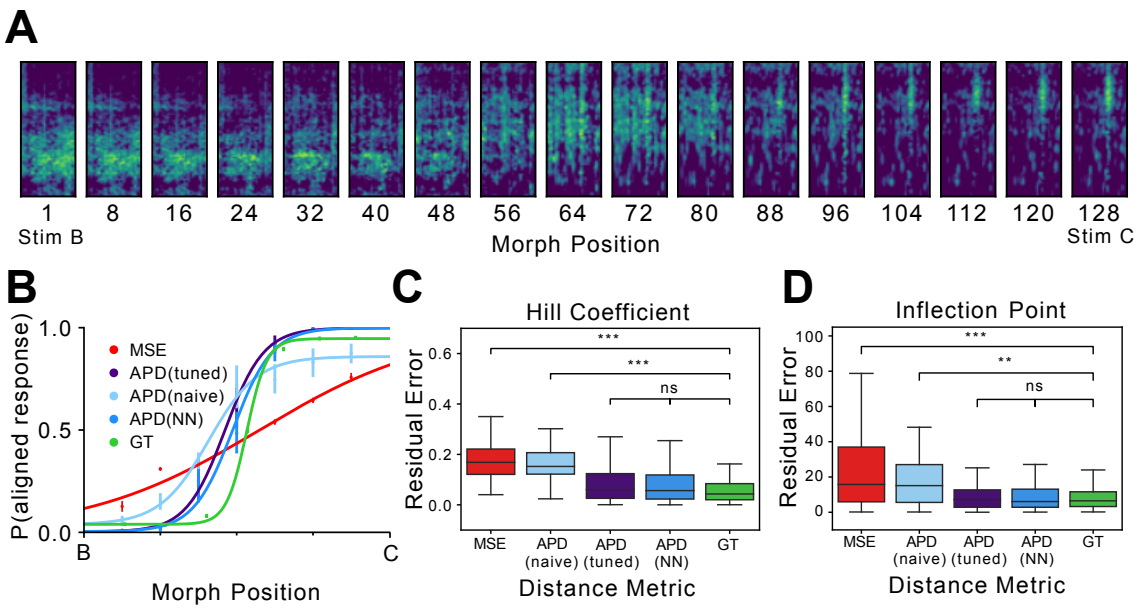

Figure 2.4 Fine-tuned APD outperforms both naive APD and MSE.

A. An example set of morph stimuli, interpolated between stimuli B (left) and C (right). Refer to Thielk, 2019 for a detailed description of stimuli generation. Briefly, the author linearly interpolated between low-dimensional representations of B and C, and reverted the interpolation vectors to spectrograms.
B. Computed and behaviorally measured psychometric curves on example morph stimuli shown in A. APD (naive) and APD (tuned) are both calculated from APD feature vectors, with the former only pre-trained (naive) and the latter fine-tuned on animal behavior data (tuned). APD (NN) refers to probability predictions made by the fine-tuned APD neural network, bypassing feature vectors.
C. Pairwise error in Hill coefficient measurements between each distance metric and the ground truth. For each computed psychometric curve (MSE, naive APD, tuned APD, and APD network predictions), we calculate the error between its computed Hill coefficient and the ground truth value under the same training conditions (morph stimuli, cohort, etc.). For the ground truth, we calculate its internal variability by measuring errors between all pairs of subject judgments under the same training conditions. Outliers are not plotted.
D. Pairwise error in inflection point measurements between distance metrics and ground truths. Error calculation follows the same pattern mentioned in C. All computed psychometric curves yield measurements within the variability of ground truths. Outliers are not plotted.

Additional comparisons between naive and tuned APD models suggest that fine-tuning is imperative to achieving perceptually accurate simulations (Fig. 2.4B-D). Without fine-tuning, both sensitivity and decision boundary accuracy yielded by naive APD drop significantly lower than the ground truth [Hill coefficient: μ=0.154, σ=0.057, $p<0.01$, LMM against GT; inflection point: μ=15.48, σ=11.97, $p<0.001$, LMM against GT] whereas sensitivity levels of the fine-tuned APD model are within the variability of the ground truth as mentioned earlier. These results indicate that accurate decision boundary placements and sensitivity characterization require fine-tuning on behavioral data.

Discussion

Our preliminary results demonstrate that the APD model achieves high fidelity in characterizing starling perception. Through the characterization of frequency sensitivity at all frequency levels, we demonstrate that the Star scale achieves much higher perceptual uniformity than other existing frequency scales such as the Hz scale and the mel scale. By measuring the accumulation of error incurred through shifting spectrograms in both the spectral and the temporal direction, we find that APD indeed addresses the instability issue common in pixel-wise errors. And finally, by incorporating behavioral data into our training pipeline, we prove that APD significantly improves sensitivity around perceptual decision boundaries to closely match that of the ground truth. For each step of training the APD model, we systematically prove that the innovative approach involved directly leads to a better performance than the existing methods and therefore contributes to the observed high fidelity.

The significance of these preliminary findings is multifaceted. First, the introduction of the Star scale answers the long-existing call for animal-specific perceptual scales. While starlings and humans share similar psychoacoustic abilities, using the Hz scale or the mel scale for research on starling perception is intrinsically problematic as neither is truly representative of starling's unique frequency sensitivity across all frequencies.

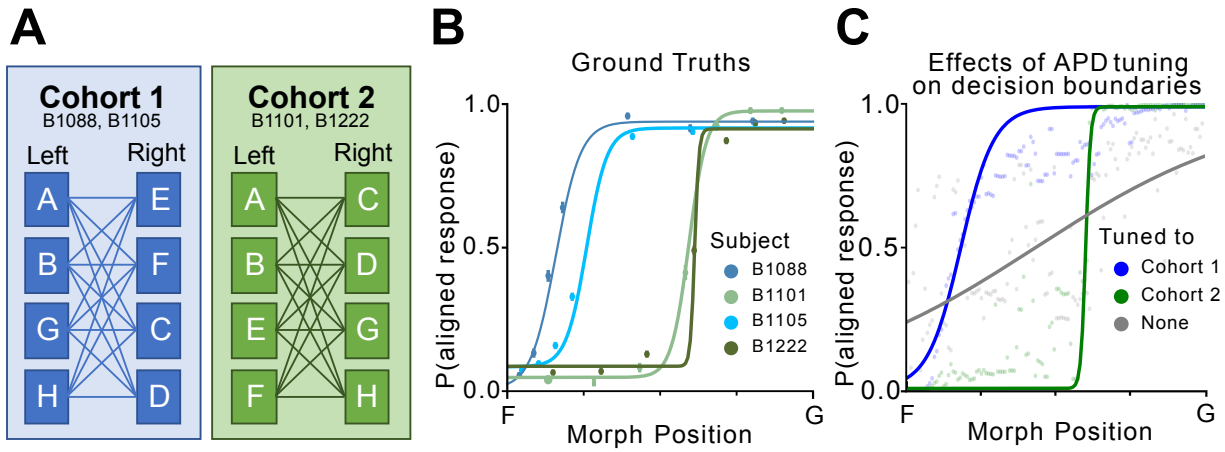

Figure 2.5 Fine-tuning realigns the model to a specific aspect of perception.

A. Training schemes of two cohorts in the original experiment. Refer to Thielk, 2019 for a detailed description of the training process. Results from cohort 1 and 2 are shown with blue hues and green hues respectively in this figure.
B. Behaviorally measured psychometric curves on morph stimuli between F and G. Each curve corresponds to a test subject in cohorts 1 and 2.
C. APD-generated psychometric curves on morph stimuli between F and G. We compare results from APD models tuned on cohorts 1 and 2, as well as APD without fine-tuning (shown in gray).

The results also suggest that the incorporation of CNN to extract perceptual embeddings is indispensable to the success of our model. Although this approach seems only natural given the abundant computer vision studies reporting success with similar approaches, our model is fundamentally distinct as all previous studies have been on visual perception. It is only through our experiments that we prove the viability of using CNN to characterize animal auditory perception. On one hand, the nature of a CNN predicate its success at extracting high-level features and consequently its robustness to local fluctuations. Our results in experiment 2 indicate that the APD not only tackles MSE's instability to small local changes but also offers different portrayals of the animal's responses to temporal and spectral shifts. Given songbirds' proven capability of processing relative pitch and relative timing (Hulse & Cynx, 1985; Rouse et al., 2021), such differential responses conform to our expectation: two audio signals differing only in the trailing or leading silence should contain resembling information whereas two signals separated by a few octaves should convey much more distinct messages. We argue the differential treatment originates from pre-training the CNN on spectrograms. Intuitively, the CNN learns to extract spectral features differently at various frequencies, but the same cannot be said about temporal features due to the sequential nature of vocalizations. In fact, if we directly use ImageNet weights, shifting in frequency domain results in APDs close to zero (Fig. 2.5), proving pre-training is essential for an accurate representation of songbirds' ability to differentially respond to temporal and spectral shifts.

Compared to universal error metrics such as MSE, a CNN-based error metric offers the unparalleled advantage of tunability. Findings in experiment 3 exemplify the significant improvement in both sensitivity characterization and decision boundary placement only made possible by fine-tuning the APD model on animal judgments. In the original behavioral experiment, subjects were divided into cohorts where training conditions differ (Fig. 2.5A). Interestingly, subjects in the same cohort arrive at similar decision boundaries on the same stimuli set while boundaries placed by different cohorts diverge (Fig. 2.5B; Thielk, 2019). We show that through differential training to mimic different cohorts, the APD model can arrive at the same diverging boundaries placed by all cohorts (Fig. 2.5C). While the naive APD fails to capture this property, its predictions between the two decision boundaries are scattered around 0.5,

indicating uncertainty, whereas the predictions beyond both boundaries are much more determined, floating close to 0 or 1. This observation offers insight into what fine-tuning does to the model: we hypothesize that fine-tuning polarizes stimuli recognition by recalibrating response probabilities for both target categories to better align with the contrasting features between targets. In other words, fine-tuning realigns the originally global feature vector to reflect specific aspects of the stimuli and subsequently assigns a recognition threshold in that feature space. As a result, what is uncertain to a naive model can be tuned in either direction depending on the training condition, leading to the observed characterization of diverging decision boundaries. Intuitively, this process matches the effect behavioral training has on experiment subjects: while the subject already has an internal measure of perceptual distance before training, the training process teaches it to focus on specific features in the signal and iteratively refines the subject's left/right decision thresholds based on feature distances.

Another advantage of the APD model is its adaptability to the user's task. Every step can be modified to fit the user's specific need: the user has not only a wide range of pretext tasks for pre-training but also unrestrained freedom to fine-tune the model. The APD model can even be expanded to species beyond starling. In the event of missing frequency sensitivity data to generate a species-specific frequency scale, the mel scale is an acceptable substitute, even showing similar results in starling perception thanks to its logarithmic nature. For starling perception, the Star scale is still recommended as it is more perceptually accurate than the mel scale.

Importantly, we believe that the APD model, while proven more perceptually accurate than existing methods, can be further optimized to achieve higher fidelity. A few components to tune include the CNN architecture, the pretext task employed during self-supervision, and the fine-tune training procedure. Our currently chosen approaches are direct adoptions from existing literature, but most have yet to be tuned for auditory perception. Another future application is to use the APD model alongside a generative neural network for spectrograms as a loss function between the target and the generated spectrograms, similar to the loss function proposed by Johnson et al. (Johnson et al., 2016).

Conclusion

In this paper, we propose the APD model, a CNN-based model to quantify the perceptual distance between two auditory signals. Training the APD model is a three-step process, for each of which we systematically prove that the innovative approach involved directly leads to better performance than the existing methods and therefore contributes to better portraying starling perception. Specifically, the APD model significantly outperforms MSE in terms of stability and sensitivity around perceptual boundaries and is therefore a more accurate representation of starling perception. In the future, we hope to optimize the APD model as well as use it as a loss function for generative neural networks for spectrograms.

Materials and Methods

Datasets

Starling Vocalization Dataset

The dataset we use for pre-training the CNN was published by Sainburg et al. and available online (Sainburg et al., 2019). It consists of songs from 14 European starlings individually collected in isolated chambers. All recordings were originally stored as 16 bit, 44.1 kHz wave files. From each singer's hour-long recordings, we randomly segment 1,500 1.4s-long continuous vocalizations. The segmentation process is done automatically so that no syllable is truncated and no motif information is taken into consideration, meaning a signal can start and end inside a motif as long as there is no continuous silence longer than 0.5s.

Morph Stimuli Dataset

The morph dataset is directly borrowed from Thielk, 2019, where a more detailed description is available. Briefly, eight arbitrarily chosen motifs (labeled A~H) are divided equally in three different ways (ABCD vs EFGH, ABGH vs EFCD, ABEF vs CDGH), forming a total of 24 unique pairs of motifs.

Spectrograms of each pair of stimuli are passed through a trained autoencoder with a 64-dimensional bottleneck. Between the two 64-dimensional latent vectors, 128 linear interpolations are extracted, reverted back to full-size spectrograms using the same autoencoder, and subsequently inverted to wave files sampled at 48kHz using methods proposed by Griffin and Lim (Griffin & Lim, 1984). Altogether, the dataset consists of 3072 morph stimuli, including repeating endpoints.

Behavioral Training Methods

Shaping

We adopt a multistage the autoshaping routine (Gentner & Hulse, 1998) that familiarizes the birds with the apparatus, guides the bird to initiate trials, and associates trials with possible food rewards. On average, it takes the subjects 3-5 days to complete shaping, after which they start behavioral trials.

Baseline Training Procedure

All subjects learn to classify natural stimuli using a two-alternative choice (2AC) procedure (Gentner & Margoliash, 2003). Each subject initiates a trial by pecking at the center port on the panel, which triggers playback of a stimuli. The subject must peck the left or right port afterwards to indicate its choice. Each stimulus is associated with a ground truth, either left or right (4 each). Incorrect responses incur punishments (timeout), while correct responses result in rewards (food access). High response rates can be achieved, and stimulus-independent response biases can be ameliorated by manipulating the reinforcement schedules or introducing remedial trials according to established procedures. The subjects are trained with a variable reinforcement ratio of 4, meaning they need to get 1-7 (average of 4) correct choices in a row to be rewarded.

Double Staircase Procedure

Once the subject is able to classify natural stimuli at a high accuracy, we start the double staircase procedure to probe the perceptual boundary between each pair of left and right natural stimuli. The procedure works by estimating a window encompassing the boundary and iteratively reducing the window edge on either side based on the subject's performance. The staircase procedure begins by randomly choosing one of the 16 possible natural stimuli pairs, and then selects a morph stimulus between the natural stimuli pair that is outside the window (90%) or just inside the window (10%). For an easy trial, the morph stimulus is one natural stimuli mixed only slightly with the other natural stimulus. For the probe trial, the stimulus is a morph just within the window the procedure believes the perceptual boundary to be in. If the subject gets a probe trial correct, the corresponding window edge advances to the location of the probe trial and further probe trials along this axis become more difficult. The subjects are rewarded by a variable reinforcement ratio mentioned above so that the birds are forced to perform on each trial but are not necessarily rewarded. This also allows for more trials per day.

Computational Methods

Star Scale

To convert a frequency value from the Hz scale to the Star scale, the following formula is used:

$$S = \begin{cases} \frac{f}{20}, & if\ f < 1600 \\ 80 + \frac{150}{\log(6.4)} \times \log\left(\frac{f}{1600}\right), & if\ f \geq 1600 \end{cases}$$

Where f is the frequency value in Hz. Note that the unit itself is meaningless, meaning it can be arbitrarily large or small depending on the multiplier attached to the formula.

Mel Scale

We use the mel scale conversion proposed by McFee et al. in Librosa (McFee et al., 2015).

Spectrogram Generation

All spectrograms used in this study are converted to the Star scale. Computationally, this is a two-step process where Hz-scale spectrograms are first constructed from sound waves using an FFT size of 2048 combined with a step size of 256. Specifically for morph stimuli, spectrograms have a low-frequency cutoff of 850 Hz and a high-frequency cutoff of 10,000 Hz to avoid silence occupying the majority of the spectrogram. The Hz-scale spectrograms are then converted to Star-scale spectrograms via a series of Star filters, which are subsequently converted to decibel-based and normalized individually. All steps other than the Star-scale conversion are accomplished with (McFee et al., 2015). The final morph spectrograms are 186 Stars tall, whereas the final vocalization spectrograms are 291 Stars tall.

Model Architecture

Briefly, the APD model consists of a CNN and a few dense layers which are only used during training and dropped for feature extraction. Specifically for results included in this study, we use VGG19 as our choice of CNN due to the abundant literature on its capability of extracting high-level features (Simonyan & Zisserman, 2014). During pre-training, we connect a single dense layer of size 4096 to the VGG19 whereas during fine-tuning, three dense layers (size 2048, 2048, 1024, respectively; ReLU activation) are used. Four separate input layers are connected simultaneously to the VGG19 during pre-training, rendering the model a Siamese network.

Pre-training

The APD model is pre-trained on a pretext task similar to the popular jigsaw task. For pre-training, we use the starling vocalization dataset in the form of Star-scale spectrograms. 90% of the dataset is assigned the training set, whereas the remaining 10% is set aside as the validation set. There is no testing set because the purpose of pre-training is to transfer learned weights and thus we are uninterested in the pretext accuracy. Each spectrogram is divided in the frequency domain into four equally sized puzzle pieces which are subsequently shuffled. The goal of the model is to reorder the puzzle pieces based on relevant information extracted. During training, all four puzzle pieces are fed into

the APD model, yielding four 512-dimensional vectors, which are then concatenated and passed to the dense layer for classification. The model is trained to minimize the MSE between target rank vectors and predicted rank vectors and is optimized with Adam (learning rate 1E-6). A batch size of 32 is used in conjunction with a maximum of 1,000 epochs and early stopping to prevent overfitting, meaning the training will stop if validation accuracy does not improve.

APD Calculation

Once the model is trained, we can calculate APD between two spectrograms. This is achieved by first dropping the dense layers and reconfiguring the model so that there is only one input layer. Both spectrograms can be passed through the model, yielding two 512-dimensional feature vectors. APD between these two spectrograms can be calculated as the cosine distance between the two vectors.

Shifting

To achieve the effect of shifting in either time or frequency domain, we pad the spectrograms with lines of silence. In the example of temporal shifting, we obtain two copies of the same spectrogram, one with n lines of silence added to the left and the other with n lines of silence added to the right. The APD model and MSE are then applied to both copies to characterize the error incurred by shifting the spectrogram.

Fine-tuning

We fine-tune the APD model on the morph stimuli dataset, under three different conditions to simulate the three cohorts in the behavioral experiment. For each set of morph stimuli judged by each subject, we train the model on all other morph stimuli sets using the subject's response probability as target. For example, to investigate the model's performance on mimicking subject X's responses to stimuli set AE, we train a model on all remaining stimuli X has been exposed to, including AF, AG, AH, etc. If during behavioral experiment X classified AF_{12} as A 95% of the time, we assign a true label of [0.95,

0.05] to the stimulus. Starting with weights from the pre-training step, the model is optimized with Adam (learning rate 1E-6) and trained to minimize categorical cross-entropy between true labels and predicted labels. To avoid overfitting, we inserted a 20% dropout after every dense layer. The output layer uses a softmax activation function to better characterize the structure of classification labels. Same as pre-training, early stopping is used to prevent overfitting.

Fitting Psychometric Curves

We model both the simulated and the measured behavior with a four-parameter logistic regression characterized by the following formula:

$$P(x) = A + \frac{K - A}{1 + e^{-B(x-M)}}$$

Where A and K are the minimum and the maximum value that can be obtained, respectively. M symbolizes the inflection point where the probability of yielding either response is 0.5. B represents Hill's coefficient, the measured slope at the inflection point.

Acknowledgement

Chapter 2, in full, in part, has been submitted for publication of the material as it may appear in PNAS 2022, Chen, Shukai; Thielk, Marvin; Gentner, Timothy Q., 2022. The dissertation author was the primary researcher and author of this paper.

References

Akbari, Hassan, Himani Arora, Liangliang Cao, and Nima Mesgarani. 2018. “Lip2audspec: Speech Reconstruction from Silent Lip Movements Video.” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. doi:10.1109/icassp.2018.8461856.

Arneodo, Ezequiel M., Shukai Chen, Daril E. Brown, Vikash Gilja, and Timothy Q. Gentner. 2021. “Neurally Driven Synthesis of Learned, Complex Vocalizations.” *Current Biology* 31 (15). doi:10.1016/j.cub.2021.05.035.

Carr, Andrew N., Quentin Berthet, Mathieu Blondel, Olivier Teboul, and Neil Zeghidour. 2021. “Self-Supervised Learning of Audio Representations from Permutations with Differentiable Ranking.” *IEEE Signal Processing Letters* 28: 708–12. doi:10.1109/lsp.2021.3067635.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. “ImageNet: A Large-Scale Hierarchical Image Database.” *2009 IEEE Conference on Computer Vision and Pattern Recognition*. doi:10.1109/cvpr.2009.5206848.

Dooling, Robert J., and Nora H. Prior. 2017. “Do We Hear What Birds Hear in Birdsong?” *Animal Behaviour* 124: 283–89. doi:10.1016/j.anbehav.2016.10.012.

Dosovitskiy, Alexey, and Thomas Brox. 2016. “Inverting Visual Representations with Convolutional Networks.” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2016.522.

Gatys, Leon A, Alexander Ecker, and Matthias Bethge. 2016. “A Neural Algorithm of Artistic Style.” *Journal of Vision* 16 (12): 326. doi:10.1167/16.12.326.

Gatys, Leon A, Alexander S Ecker, and Matthias Bethge. 2017. “Texture and Art with Deep Neural Networks.” *Current Opinion in Neurobiology* 46: 178–86. doi:10.1016/j.conb.2017.08.019.

Gentner, Timothy Q, and Stewart H Hulse. 1998. “Perceptual Mechanisms for Individual Vocal Recognition in European Starlings,Sturnus Vulgaris.” *Animal Behaviour* 56 (3): 579–94. doi:10.1006/anbe.1998.0810.

Gentner, Timothy Q., and Daniel Margoliash. 2003. “Neuronal Populations and Single Cells Representing Learned Auditory Objects.” *Nature* 424 (6949): 669–74. doi:10.1038/nature01731.

Griffin, D., and Jae Lim. 1984. “Signal Estimation from Modified Short-Time Fourier Transform.” *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32 (2): 236–43. doi:10.1109/tassp.1984.1164317.

Hill, Archibald Vivian. 1913. “The Combinations of Haemoglobin with Oxygen and with Carbon Monoxide. I.” *Biochemical Journal* 7 (5): 471–80. doi:10.1042/bj0070471.

Hill, AV. 1910. “The Possible Effects of the Aggregation of the Molecules of Haemoglobin on Its Oxygen Dissociation.” *Proceedings of the Physiological Society*, January.

Hulse, Stewart H., and Jeffrey Cynx. 1985. “Relative Pitch Perception Is Constrained by Absolute Pitch in Songbirds (Mimus, Molothrus, and Sturnus).” *Journal of Comparative Psychology* 99 (2): 176–96. doi:10.1037/0735-7036.99.2.176.

Jing, Longlong, and Yingli Tian. 2021. “Self-Supervised Visual Feature Learning with Deep Neural Networks: A Survey.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (11): 4037–58. doi:10.1109/tpami.2020.2992393.

Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. 2016. “Perceptual Losses for Real-Time Style Transfer and Super-Resolution.” *Computer Vision – ECCV 2016*, 694–711. doi:10.1007/978-3-319-46475-6_43.

Klump, Georg M., and Elke H. Maier. 1989. “Gap Detection in the Starling (Sturnus Vulgaris).” *Journal of Comparative Physiology A* 164 (4): 531–38. doi:10.1007/bf00610446.

Kolesnikov, Alexander, Xiaohua Zhai, and Lucas Beyer. 2019. “Revisiting Self-Supervised Visual Representation Learning.” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2019.00202.

Kornblith, Simon, Jonathon Shlens, and Quoc V. Le. 2019. “Do Better Imagenet Models Transfer Better?” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2019.00277.

Kuhn, A., H. -J. Leppelsack, and J. Schwartzkopff. 1980. “Measurement of Frequency Discrimination in the Starling (Sturnus Vulgaris) by Conditioning of Heart Rate.” *Naturwissenschaften* 67 (2): 102–3. doi:10.1007/bf01054703.

Mahendran, Aravindh, and Andrea Vedaldi. 2015. “Understanding Deep Image Representations by Inverting Them.” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2015.7299155.

McFee, Brian, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. “Librosa: Audio and Music Signal Analysis in Python.” *Proceedings of the 14th Python in Science Conference*. doi:10.25080/majora-7b98e3ed-003.

Noroozi, Mehdi, and Paolo Favaro. 2016. “Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles.” *Computer Vision – ECCV 2016*, 69–84. doi:10.1007/978-3-319-46466-4_5.

Purwins, Hendrik, Bo Li, Tuomas Virtanen, Jan Schluter, Shuo-Yiin Chang, and Tara Sainath. 2019. “Deep Learning for Audio Signal Processing.” *IEEE Journal of Selected Topics in Signal Processing* 13 (2): 206–19. doi:10.1109/jstsp.2019.2908700.

Rouse, Andrew A., Aniruddh D. Patel, and Mimi H. Kao. 2021. “Vocal Learning and Flexible Rhythm Pattern Perception Are Linked: Evidence from Songbirds.” *Proceedings of the National Academy of Sciences* 118 (29). doi:10.1073/pnas.2026130118.

Sainburg, Tim, Brad Theilman, Marvin Thielk, and Timothy Q. Gentner. 2019. “Parallels in the Sequential Organization of Birdsong and Human Speech.” *Nature Communications* 10 (1). doi:10.1038/s41467-019-11605-y.

Simonyan, Karen, and Andrew Zisserman. 2014. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” *ArXiv Preprint ArXiv:1409.1556*.

Stevens, S. S., J. Volkmann, and E. B. Newman. 1937. “A Scale for the Measurement of the Psychological Magnitude Pitch.” *The Journal of the Acoustical Society of America* 8 (3): 185–90. doi:10.1121/1.1915893.

Thielk, Marvin. 2019. “The Unreasonable Effectiveness of Machine Learning in Neuroscience: Understanding High-Dimensional Neural Representations with Realistic Synthetic Stimuli.” Thesis, La Jolla: UC San Diego Electronic Theses and Dissertations. UC San Diego.

Weiss, James N. 1997. “The Hill Equation Revisited: Uses and Misuses.” *The FASEB Journal* 11 (11): 835–41. doi:10.1096/fasebj.11.11.9285481.

Zhai, Xiaohua, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. 2019. “S4L: Self-Supervised Semi-Supervised Learning.” *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. doi:10.1109/iccv.2019.00156.

CHAPTER 3

Abstract

Songbird vocalizations are complex in form and in function: short bursts of calls and long continuous songs, while serving different purposes in each species, are both prevalent in birdsongs. Some species, such as European starlings, utilize vocalizations for singer recognition, mainly through memorizing the organization of a familiar singer's unique song components. However, recent studies reveal that European starlings are similar to humans in that they are capable of identifying singers based on sub-syllable level acoustic features, a vocal signature. In this study, we explore the possibility of starlings using sound textures as a vocal signature to identify familiar singers. We first prove that a subject's song textures converge to a stable level by calculating the cosine similarities between short and long segments of vocalizations. We then demonstrate the strong correlation between texture clustering and singer identities, using both mutual information and neural networks. Finally, we show through behavioral experiments that starlings previously trained on singer recognition can also classify noise-like synthetic signals from their familiar textures.

Introduction

Songbirds, like humans, communicate through vocal signals that are complex in the form: short bursts of calls and long continuous vocalizations are both prevalent in birdsongs. Apart from their structural complexity, these signals encode various information specific to the species or even the individual's repertoire, such as hunger, signals of danger, or interests in mating.

One crucial piece of such embedded information is individual identity. While the ability to recognize familiar individuals from their vocalizations is observed in many social animals including birds

and mammals (Gentner, 2008, Vignal et al., 2004, Charrier et al., 2001), different species capitalize on distinct vocal cues for individual identification.

An interesting case is European starlings. They rely mainly on syntactic cues for recognizing familiar conspecifics, specifically by memorizing the organization of their unique song components (Gentner & Hulse, 2000). However, syntactic cues are not present in all vocalizations where subject recognition happens. In fact, starlings' individual identification can still be achieved when syntactic cues are artificially removed (Gentner, 2008). Several lines of evidence point to the presence of certain auditory features that aid in starling conspecific recognition, a vocal signature.

Some vocal features have been proposed as viable vocal signatures in human speaker recognition, such as Mel-frequency cepstral coefficients (MFCCs); however, MFCC's noise robustness is costly to compensate (Zhao & Wang, 2013). In this paper, we focus on texture statistics proposed by McDermott and Simoncelli, and propose vocal textures as a set of acoustic features for subject identification in European starlings.

A texture is defined as characteristics that remain constant, originally referring to tactual properties such as roughness, hardness, etc.. For the past few decades, studies on textures have expanded to visual textures (Julesz, 1962) and sound textures (McDermott & Simoncelli, 2011). Conventionally, sound textures represent sounds that are not strictly stationary but exhibit semi-stationary characteristics, such as applause, wind, or crackling log fire. Studies show that human listeners can accurately categorize various kinds of sound textures, even synthetic ones with temporal homogeneities matching those of natural textures, suggesting the involvement of time-averaged features in human auditory perception (McDermott & Simoncelli, 2011). However, many questions remain unanswered. Can texture be used to describe less homogeneous sounds such as vocalizations? If so, what information do textures encode in the auditory perception of nonhuman animals?

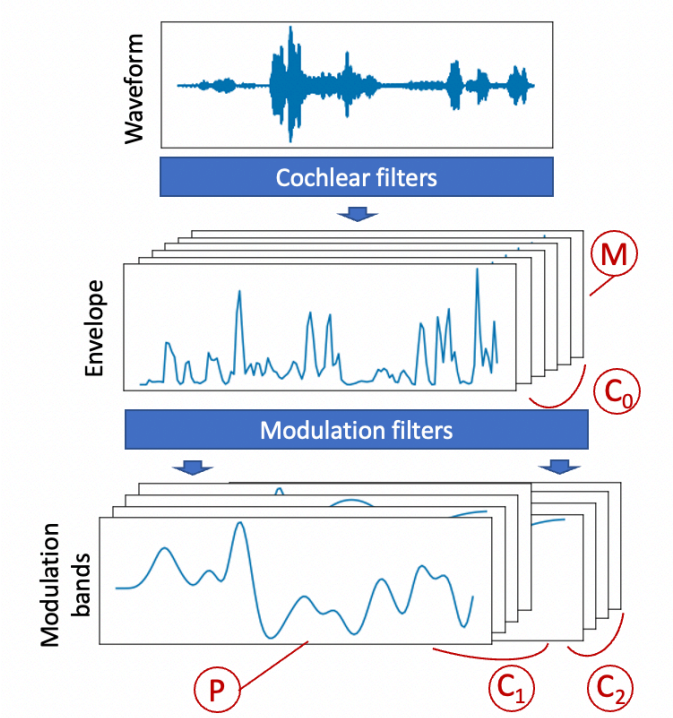

Figure 3.1 Sound texture computation

For a more detailed description of the algorithm, refer to McDermott & Simoncelli, 2011. Briefly, a sound pressure waveform is first passed through a set of cochlear filters. The envelopes of the filtered sub-bands are collected, from which marginal statistics (M) and correlations (C0) are computed. Each envelope is then filtered by a set of modulation filters, yielding modulation bands. We then calculate the modulation power (P), inter-modulation band correlations (C1, referring to correlations between modulation bands from different envelopes), and intra-modulation band correlations (C2, referring to correlations between modulation bands from the same envelope).

In this study, we explore the existence of subject-specific vocal textures and their viability as a vocal signature in European starlings. We use summary statistics, proposed by McDermott and Simoncelli, to characterize sound textures (Fig. 1). First, we investigated whether vocal signals of various lengths from the same singer can converge to a stable texture. Next, we visualize the distribution of vocal textures across several subjects and compute the amount of subject information embedded in each and all summary statistics. Lastly, we test starlings with noise-like synthetic signals that are embedded with familiar textures and evaluate their ability to generalize on familiar textures.

Terminology

Researchers studying different animals often have varied terminology to describe similar concepts in animal vocalizations. For consistency, we conduct our analyses using two hierarchical units of continuous vocalizations, "syllable" and "bout", the terms and definitions of which are following. A "syllable" is a unit of sound separated by silence, usually consisting of one or more notes which are defined as continuous markings on a spectrogram. A "bout" is a continuous sequence of syllables, with short silences in between; a bout ends at the syllable uttered before an extended pause. For many species of songbirds, a bout consists of several motifs, an intermediate unit composed of multiple syllables and observed repeatedly. However, we choose not to include motifs in the scope of this paper as this paper focuses primarily on sub-syllable level vocal features rather than syntactic cues where the concept of motifs plays a crucial role.

Results

For analyses in this study, we select the same set of texture components chosen by McDermott & Simoncelli: envelope marginals (M, including mean, standard deviation, skewness, and kurtosis), envelope correlations (C0), modulation power (P), inter-modulation band correlations (C1), and intra-modulation band correlations (C2).

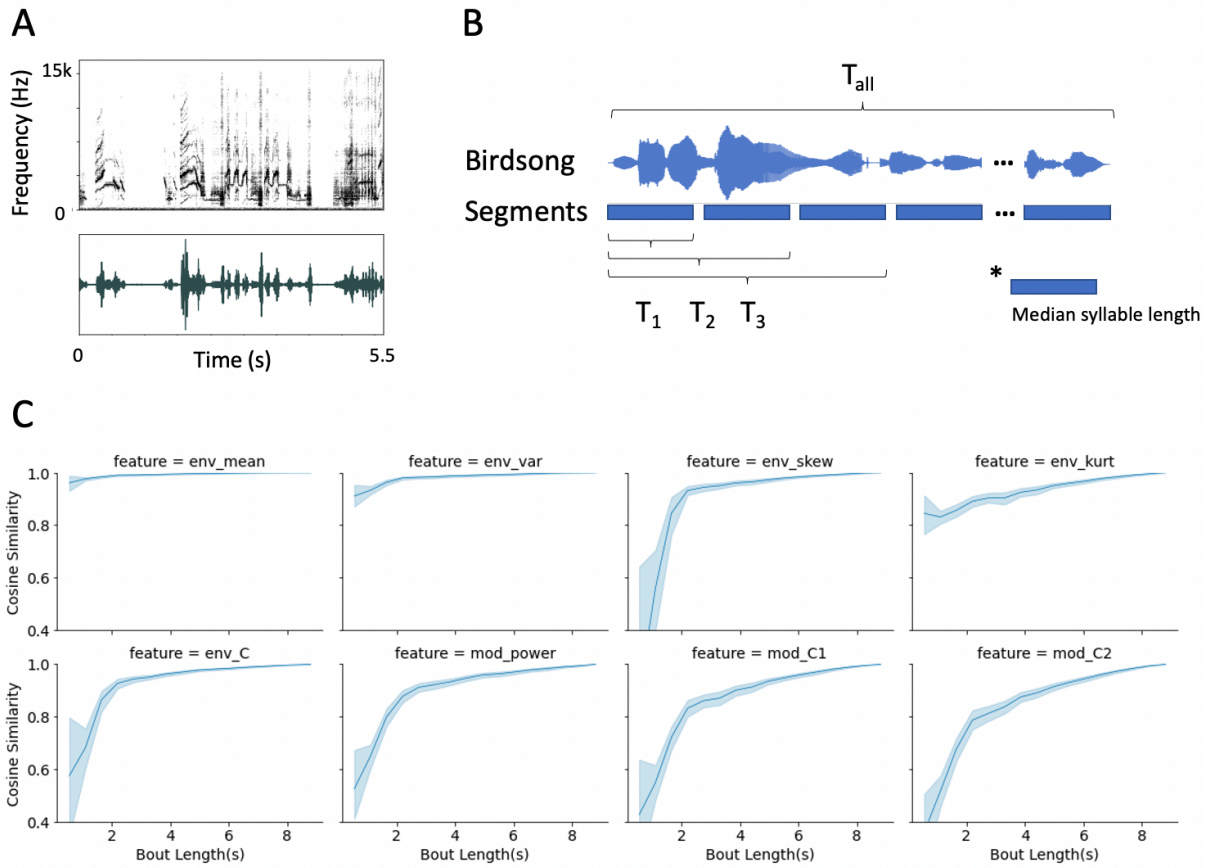

Figure 3.2 Starling song texture stabilizes and converges as the singer continues to vocalize.

A) Example spectrogram and pressure waveform of a European starling song.
B) Segmentation scheme used to determine stable bout length. A continuous vocalization of a species is segmented into increasing lengths; each unit of length is the measured median syllable length of that species.
C) We plot the cosine similarity between short segment texture and full-length bout texture. All starling texture components stabilize as vocalization lengths increase.

Texture Convergence

For vocal texture to be used as a vocal signature for subject identification, one premise is for a subject's texture to stabilize and converge as the bird continues to vocalize. While we expect the texture to change rapidly in short time frames due to the varying nature of rhythms and tones within vocalizations, we theorize that eventually, the texture becomes more stable as rapid variations of rhythms and tones get averaged over longer periods of time.

To test our theory, we devise a segmentation scheme and operate it on on a large dataset of starling vocalizations collected from 14 subjects (Sainburg et al., 2019, Fig. 3.2B). 100 non-overlapping bouts are randomly selected from each subject's repertoire. Every bout yields 80 segments of increasing length, each incrementing by approximately 110ms, the computed median syllable length, from the previous. We choose to increment by the median syllable length in an effort to easily transfer to other species' vocalizations. We subsequently extract vocal texture from all 112,000 segments of vocalizations, and compute the textural similarity between each pair of segment texture and its corresponding full-length bout texture.

As shown in Fig. 3.2C, it is clear the results conform to our expectations. Across all texture components, a similar trend is observed: while the textural similarity between a full-length bout and a few syllables segmented from it is relatively low and volatile, the similarity quickly rises as the number of syllables grows, and eventually converges to 100%. Therefore, we define the textually stable length for starling songs as the minimum length of continuous singing to reach 90% bout texture on all texture components. The textually stable length as measured from 112,000 segments is approximately 4.4s.

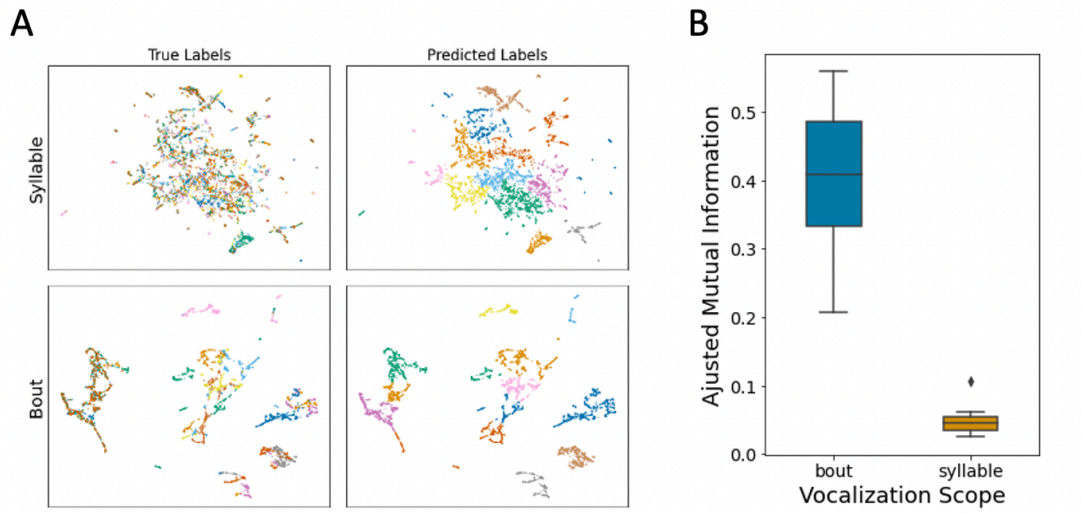

Figure 3.3 Quantifying subject information embedded in vocal texture through clustering and mutual information.

A. Example distribution of true labels and predicted labels of European starling vocalizations in UMAP space. For this example visualization, both syllable and bout envelope means are shown. Predicted labels are generated using agglomerative clustering. Each color represents a different label.
B. Adjusted mutual information between true and predicted labels of starling songs across all texture components.

Quantifying Subject Information Embedded in Vocal Texture

For song texture to qualify as a vocal signature, we need to determine the amount of singer information embedded within. Inspired by the results from our previous analysis, we hypothesize that short bursts of vocal signals carry little to no amount of singer identity whereas long continuous bouts reveal much more information about the singer. To verify these hypotheses, we segmented the same starling dataset into vocal signals of two different lengths: 7,000 single syllables (about 110ms long) and 7,000 textually stable bouts (about 4.4s long). Additionally, we adopted two computational approaches: one unsupervised approach through clustering as well as one supervised method using neural networks.

The unsupervised approach is a two step process. We first reduce all texture components to two dimensions through UMAP, a dimension reduction algorithm often used for visualization that can reduce high-dimensional data to as low as 2D while preserving its global structure. Next, we apply a clustering algorithm to UMAP-reduced 2D data, and assign predicted labels to the identified clusters. Here we use agglomerative clustering because of its hierarchical nature and because there is no need to pre-specify the number of clusters. Lastly, We calculate the adjusted mutual information (AMI) between all true labels, meaning singer identities, and all predicted labels, in other words cluster labels determined by our algorithm. We choose AMI because it is normalized, meaning an AMI of zero corresponds to chance.

A brief glance at the distribution of UMAP-reduced data (Fig. 3.3A) tells us that bout textures form much clearer clusters than syllable textures, confirming our expectation. In addition, bout textures yield easily identifiable clusters that positively correlate with singer labels. This observation is confirmed by the AMI between true labels and predicted labels (Fig. 3.3B): bout textures yield significantly positive values (mean=0.398, standard deviation=0.125, range=[0.207, 0.559]), suggesting a high level of singer information embedded in bouts. Interestingly, while syllable textures result in AMI that is much closer to chance than bout textures (mean=0.051, standard deviation=0.025, range=[0.026, 0.106]), all texture components still yield a positive AMI value, suggesting weak, albeit still present, singer information.

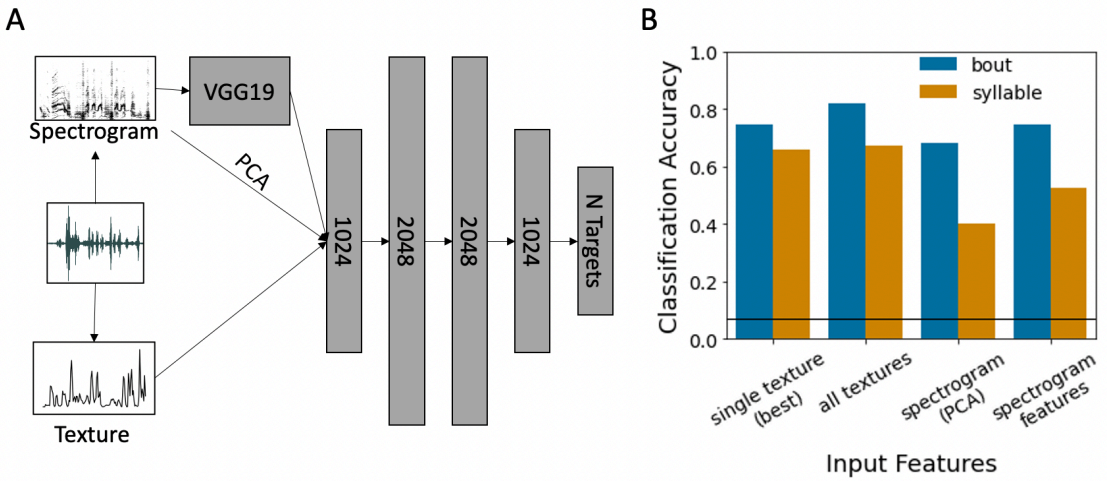

Figure 3.4 Quantifying subject information embedded in vocal texture through neural network classification.

A. Architecture used for subject classification. Texture statistics are used directly as inputs for the FFNN, whereas features extracted with a VGG network and PCA are used as inputs for spectrograms.

B. Classification accuracies with various input features extracted from vocal signals. Other than single texture classification accuracies, classification accuracies with all textures combined and spectrogram classification accuracies (with and without feature extraction) are also shown.

In parallel, the supervised method involves a neural network that takes texture data as inputs, with the goal of predicting singer labels (Fig. 3.4A). Detailed training parameters are documented in Materials and Methods. Briefly, we train the network on 80% of each texture dataset, and test on the rest 20%. One hypothesis we make is that more than all texture components combined contain more singer information than a single texture. In order to test this hypothesis, we inspect the classification accuracy of every texture component as well as all components combined. In addition, to evaluate the effectiveness of texture classification, we train two separate networks in parallel, on top 512 principal components of corresponding spectrograms, and 512 features extracted from the spectrograms with a convolutional network (CNN) respectively.

A few conclusions can be drawn from classification accuracies shown in Fig. 3.4B. Firstly, envelope mean, the best-performing single texture component, outclasses both principal components and features extracted from spectrograms. In fact, a few other texture components also reach or surpass the spectrogram predictions (Fig. XXX). Moreover, comparisons between single texture and all textures combined confirm our hypothesis that the latter carry more singer information than any single texture component, on both syllable and bout texture datasets. Lastly, syllable textures achieved surprisingly high classification accuracies when the network is trained on either a single texture (65.9%) or all textures combined (67.3%). Both are more than 10% higher than that achieved by spectrogram features (52.3%), suggesting that song textures are better features for extracting singer information from short bursts of vocalizations than spectrograms.

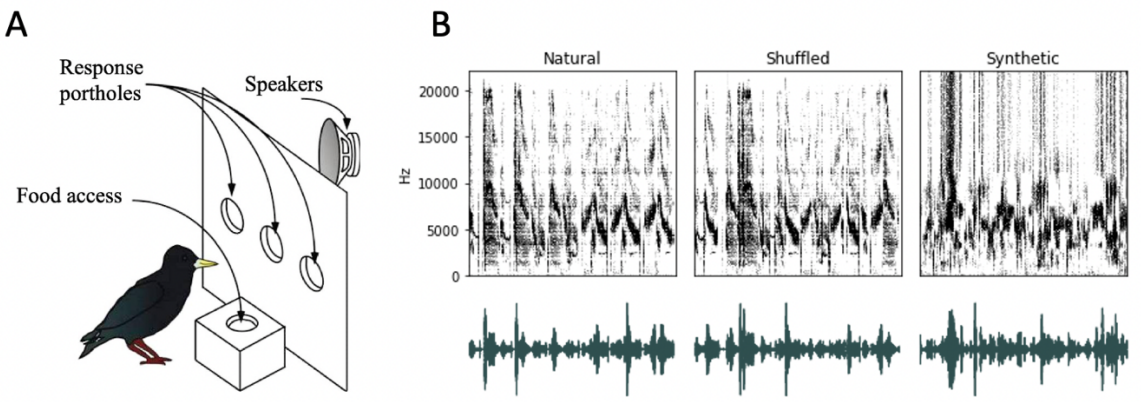

Figure 3.5 Behavior experiment setup.

A. A diagram of the operant apparatus that consists of three response ports with infrared detection sensors, a food hopper on the bottom to provide access to a food reward, and a speaker hidden behind the panel that plays the stimuli when a trial is initiated.
B. Example spectrograms and waveforms of stimuli used in the experiment. The shuffled stimulus is generated directly from the natural stimulus on the left, whereas the synthetic stimulus is synthesized from the shuffled stimulus. The example synthetic stimulus include all texture components.

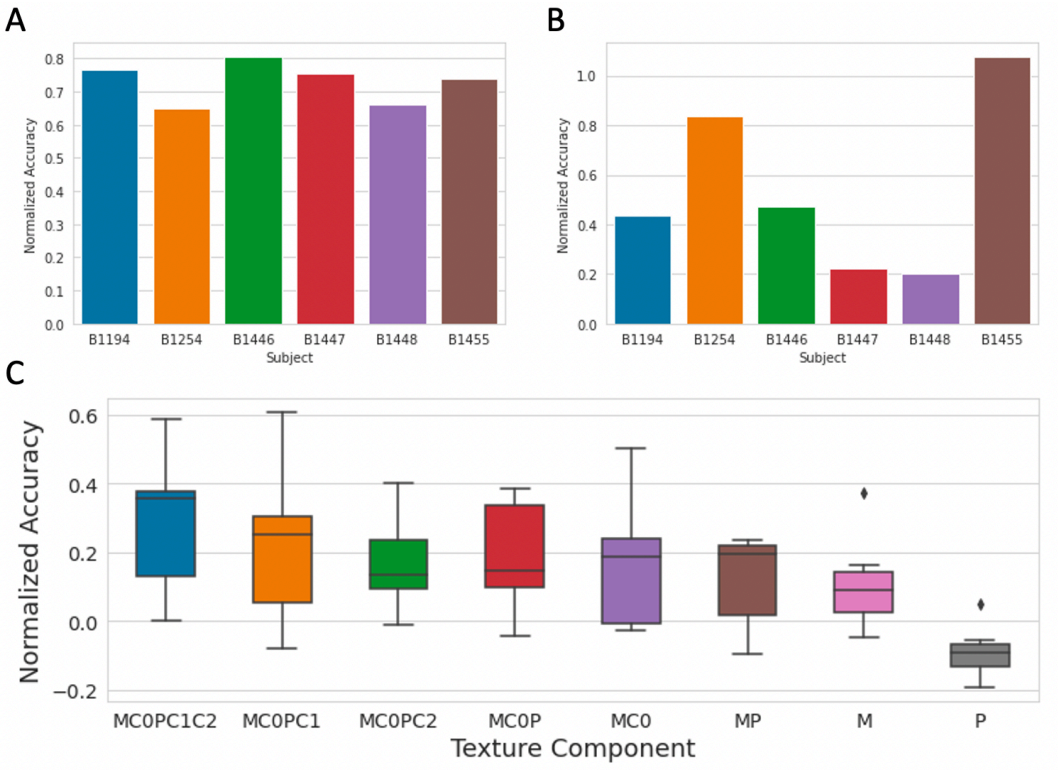

Figure 3.6 Behavior experiment results.

A. Normalized recognition accuracy on natural novel songs from familiar singers.
B. Normalized recognition accuracy on shuffled novel songs from familiar singers.
C. Normalized recognition accuracy on synthetic stimuli that are embedded with familiar texture. Results from all test subjects are combined, with whiskers showing 1.5 times interquartile range (IQR).

Behavioral Analyses

Results from computational analyses above confirm our intuition that song textures carry significant singer information and therefore can potentially be used as a vocal signature. However, it remains unclear whether starlings actively use them for singer identification.

To answer this question, we design a behavior experiment where we train starlings on singer recognition and then probe them with noise-like synthetic signals that are embedded with familiar textures. More specifically, six subjects are included in this study, each trained in a two-alternative choice (2AC) experiment to recognize a pair of singers through a large repertoire of natural songs (Fig. 3.5A). Once a subject's recognition accuracy stabilizes to a high level, we probe it with a series of novel stimuli and normalize test accuracies between training performance and chance. This means a normalized accuracy of 100% is on the same level as familiar song recognition, whereas a normalized accuracy score of 0% means recognition is around chance. Such normalization is based on our intuition that novel song recognition should not exceed familiar song recognition.

The first set of probe signals are novel songs from familiar singers, the goal of which is to evaluate the bird's ability to generalize on singer identity. As seen in Fig. 3.6A, all subjects are able to achieve a significantly higher than chance normalized recognition accuracy (65%~80%) on novel songs, indicating the subjects achieving singer identity generalization.

Next, we asks the subject to respond to shuffled novel songs from familiar singers, similar to the experiment conducted by Gentner (Gentner, 2008). These shuffled songs are void of syntactic cues so any singer identification is entirely driven by vocal feature recognition. Although the recognition accuracies on shuffled songs vary significantly among subjects (Fig. 3.6B, 19%~105%), all subjects are able to classify shuffled songs reliably, hinting at their ability to recognize a familiar singer from their vocal signature.

Finally, we probe the subjects with texture stimuli that are artificially generated from noise, albeit matching the texture of songs from their familiar singers. We choose to generate signals from noise to

ensure no song-like structure is present in the synthetic signals, so any recognition is solely driven by texture recognition. In addition, starting with all texture components combined, we follow the experimental paradigm in McDermott & Simoncelli by dropping one texture components at a time when synthesizing texture stimuli, until only envelope marginals or modulation power is left. A few observations can be made from the titration results (Fig. 3.6C). First and foremost, all but one test condition lead to higher than chance recognition accuracy, proving starlings' capability of identifying familiar singers from their song texture. Moreover, sequentially dropping texture components results in a decreasing trend in recognition accuracies, suggesting multiple texture components are utilized in singer recognition. Lastly, not all subjects successfully recognize the "singer" from texture. Over time, one of our subjects has developed a strong side bias whenever synthetic signals are played, meaning they only choose the same one side regardless of the "singer". There are a few explanations for this observation, the most likely being the bird simply employing its own strategy on any unfamiliar stimulus instead of transferring knowledge from its training stimuli. There is also a possibility that this specific bird is incapable of song texture recognition, because in past experiences subjects are more prone to developing a side bias over difficult tasks. This would mean singer identification through song texture, while achievable, is not universal in European starlings.

Future Research

From the very beginning, we have generalizability in mind and develop the pipeline to be easily transferable to other species. In fact, our preliminary research show that the clustering observed in starling songs is also present in other songbirds, mammals and even humans. Similarly, neural networks trained to classify vocal textures from these species also achieve accuracies that are significantly higher than chance. All evidence points to a promising avenue for researchers working on other species to pursue.

Meanwhile, we plan to continue our research on European starling through chronic physiology experiments where trained subjects will be implanted with a silicone probe in NCM, the auditory cortex

in songbirds. We will record neural spiking activities when the subject does trials, and attempt to identify similarities in neural activities responding to recognizable natural songs and recognizable artificial textures, as well as dissimilarities between recognizable and unrecognizable textures.

Conclusion

In this study, we explore the possibility of starlings using sound textures as a vocal signature to identify familiar singers. We first prove that a subject's song textures converge to a stable level by calculating the textual similarities between short and long segments of vocalizations. We then demonstrate the abundant singer information embedded in song texture, using both mutual information and neural networks. Finally, we show through behavioral experiments that singer identification through song texture is achievable in European starlings. Our pipeline is easily transferable to other species, with preliminary results pointing to a promising avenue for researchers working on other species to pursue.

Materials and Methods

Datasets

The starling vocalization dataset we use for this study was published by Sainburg et al. and available online (Sainburg et al., 2019). It consists of songs from 14 European starlings individually collected in isolated chambers. All recordings were originally stored as 16 bit, 44.1 kHz wave files.

Data Curation

All segmentation processes in this study are done automatically without truncating any syllables. This is achieved by first plotting the envelope of the vocal signal through downsampling, setting a 50ms tolerance on both the start and the end of every target segment, and only confirming a successful segment if both fall within the valleys in the envelope.

Starling Syllables

We extract all syllables from the entire dataset and calculate the median syllable length (110ms). We randomly select 500 syllables from each singer's recording to form a syllable dataset for the computational analyses. In total, there are 7,000 starling syllable segments that are approximately 110ms long.

Texture Convergence Analysis

From each singer's hour-long recordings, we randomly segment 100 nonoverlapping 8.8s-long continuous vocalizations, referred to as "full-length bouts". We subsequently truncate each full-length bout into 80 segments of increasing length, starting from the start of the bout, with each following segment incrementing by approximately 110ms.

Textually Stable Starling Bouts

Once the textually stable starling song length is set to 4.4s, we randomly segment 500 nonoverlapping 4.4s-long bouts from each singer's recordings, referred to as "stable bouts". These bouts are used for both the computational analyses and the behavioral experiment (referred to as "natural stimuli"). In total, there are 7,000 starling stable bout segments that are approximately 4.4s long.

Shuffling Stable Bouts

From the 500 stable bouts per singer, we randomly choose 20 and segment down to individual syllables, the order of which is then shuffled to form the shuffle dataset for the behavior experiment, referred to as "shuffled stimuli".

Generating Synthetic Stimuli

From the 20 stable bouts per singer, we randomly select five and synthesize a set of texture-matched synthetic stimuli from noise using an algorithm developed by McDermott & Simoncelli (McDermott & Simoncelli, 2011). Briefly, the synthesis is achieved by iteratively modifying the noise

and matching its texture to the stable bout texture. For each stable bout, we generate a set of eight stimuli matching the following texture components respectively:

1. Envelope marginals (M)
2. Modulation power (P)
3. Envelope marginals+envelope correlations (MC0)
4. Envelope marginals+modulation power (MP)
5. Envelope marginals+modulation power+envelope correlations (MC0P)
6. Envelope marginals+modulation power+envelope correlations+inter-modulation band correlations (MC0PC1)
7. Envelope marginals+modulation power+envelope correlations+intra-modulation band correlations (MC0PC2)
8. Envelope marginals+modulation power+envelope correlations+intra-modulation band correlations+inter-modulation band correlations (MC0PC1C2)

Clustering

Dimensionality Reduction

We reduce the dimenaionality of each texture component to 2D using UMAP (McInnes et al., 2018). Here are the parameters we use for UMAP: the number of neighbors is set to 20; the number of components is set to 2; the minimum distance is set to 0.1, the default value; the metric we use is "cosine" which refers to the cosine distance function.

Clustering Algorithm

Here we use agglomerative clustering because of its hierarchical nature and because there is no need to pre-specify the number of clusters. We apply the agglomerative clustering function from scikit learn directly on the UMAP-reduced 2D texture data, with affinity set to "euclidean" (Pedregosa et al., 2011).

Clusterability Analysis

We adopt the adjusted mutual information function from scikit learn, based on the concept first proposed by Vinh et al. (Pedregosa et al., 2011, Vinh et al., 2009). All parameters are set to default.

Neural Networks

Architecture

As shown in Fig 3.4A, a feed forward neural network (FFNN) with 4 hidden layers is used in this study. Depending on the specific test case, the network can be supplied with a range of inputs including texture data (both single component and all components combined), principal components of starling spectrograms (512 dimensions), and features extracted from starling spectrograms using a VGG19 network loaded with ImageNet weights (512 dimensions, Simonyan & Zisserman, 2015, Deng et al., 2009). A 20% dropout is implemented to every dense layer to prevent overfitting. Because the targets are one-hot encoded, we use softmax as the activation function for the output layer.

Training Scheme

For both the syllable dataset and the stable bout dataset, we randomly divide the entire dataset into five 20% subsets. Iteratively, we select one subset for testing and the other four for training. Effectively, we train the network on 80% of the entire dataset and test on the rest 20%; once the training is complete, we iterate to the next 20% subset.

Since the targets are one-hot encoded singer labels, the network is trained to minimize the categorical crossentropy between targets and outputs, using Adam as the optimizer (Kingma & Ba, 2015). To further mitigate overfitting, we incorporate early stopping in our training scheme, where training stops if the validation loss increases between epochs. The maximum number of training epochs without early stopping is set to 500.

Behavior Training

Subjects

Six European starlings, captured from the wild as adults were used in this experiment. All of the birds were naive to operant experimental procedures. We do not control the sex of the subjects. They are housed in a large, mixed-sex, conspecific aviary with ad libitum access to food and water from the time of capture until being moved into the testing chamber. The lights in the aviary follow the schedule of local sunrise and sunset.

Shaping

We adopt a multistage autoshaping routine that familiarizes the birds with the apparatus, guides the bird to initiate trials, and associates trials with possible food rewards (Gentner & Hulse, 1998). On average, it takes the subjects 3-5 days to complete shaping, after which they start behavioral trials.

Two Alternative Choice Experiment

All subjects learn to classify natural stimuli using a two-alternative choice (2AC) procedure (Gentner & Margoliash, 2003). Each subject initiates a trial by pecking at the center port on the panel, which triggers playback of a stimuli. The subject must peck the left or right port afterwards to indicate its choice. Each stimulus is associated with a ground truth, either left or right, in the case of our experiment associated with a singer. Incorrect responses incur punishments (timeout), while correct responses result in rewards (food access). High response rates can be achieved, and stimulus-independent response biases can be ameliorated by manipulating the reinforcement schedules or introducing remedial trials according to established procedures. The subjects start with a fixed ratio of 1, meaning every time they make the correct response they get rewarded. They also start with two natural stimuli on each side, and whenever the recognition accuracy reaches 80%, we double the number of stimuli on both sides. The process repeats until the recognition accuracy on novel natural stimuli plateau. At that point, we progressively switch to a variable reinforcement ratio of 2.5, meaning the subjects need to get 1-4 (average of 2.5) correct choices in a row to be rewarded.

Probing Procedure

The variable reinforcement schedule enables us to insert probe stimuli at the beginning of a variable reinforcement chain without having to reinforce them. At this point, we sequentially probe the subjects with novel natural stimuli, shuffled stimuli, and synthetic stimuli, all from each subject's familiar singers. The subjects do not get punished or rewarded for any choice they make on the probe stimuli. We implement forced choice on probe stimuli, meaning if the bird ignores a probe stimulus, the same stimulus gets played next time a trial is initiated, until the bird finally makes a choice. Once the bird respond to the probe stimulus, a normal variable reinforcement chain of familiar natural stimuli starts.

Acknowledgement

Chapter 3, in part, is currently being prepared for submission for publication of the material. Chen, Shukai; Gentner, Timothy Q. The dissertation author was the primary researcher and author of this material.

References

Charrier I, Mathevon N, Jouventin P. Mother's voice recognition by seal pups. Nature. 2001;412(6850):873–.

Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009;

Gentner Timothy Q, Hulse Stewart H. Perceptual mechanisms for individual vocal recognition in European starlings,Sturnus vulgaris. Animal Behaviour. 1998;56(3):579–94.

Gentner TQ, Hulse SH. Perceptual classification based on the component structure of song in European Starlings. The Journal of the Acoustical Society of America. 2000;107(6):3369–81.

Gentner TQ, Margoliash D. Neuronal populations and single cells representing learned auditory objects. Nature. 2003;424(6949):669–74.

Gentner TQ. Temporal scales of auditory objects underlying birdsong vocal recognition. The Journal of the Acoustical Society of America. 2008;124(2):1350–9.

Julesz B. Visual pattern discrimination. IEEE Transactions on Information Theory. 1962;8(2):84–92.

Kingma D, Ba J. Adam: A method for stochastic optimization [Internet]. arXiv.org. 2015 [cited 2022Aug3]. Available from: https://arxiv.org/abs/1412.6980v8

McDermott JH, Simoncelli EP. Sound texture perception Via statistics of the AUDITORY PERIPHERY: Evidence from sound synthesis. Neuron. 2011;71(5):926–40.

McInnes L, Healy J, Saul N, Großberger L. UMAP: Uniform manifold approximation and projection. Journal of Open Source Software. 2018;3(29):861.

Pedregosa F, Profile V, Varoquaux G, Gramfort A, Michel V, Thirion B, et al. Scikit-Learn: Machine learning in Python [Internet]. The Journal of Machine Learning Research. 2011 [cited 2022Aug3]. Available from: https://dl.acm.org/doi/10.5555/1953048.2078195

Sainburg T, Theilman B, Thielk M, Gentner TQ. Parallels in the Sequential Organization of Birdsong and Human speech. Nature Communications. 2019;10(1).

Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [Internet]. arXiv.org. 2015 [cited 2022Aug3]. Available from: https://arxiv.org/abs/1409.1556

Vignal C, Mathevon N, Mottin S. Audience drives male songbird response to partner's voice. Nature. 2004;430(6998):448–51.

Vinh NX, Epps J, Bailey J. Information theoretic measures for clusterings comparison. Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09. 2009;

Zhao X, Wang DL. Analyzing noise robustness of MFCC and GFCC features in speaker identification. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. 2013;

CONCLUSION

When I first started my Ph.D. in 2015, machine learning, especially deep neural networks, was experiencing exponential growth. With the introduction of unsupervised techniques like generative adversarial networks (GANs), self-supervision, etc., researchers in all fields started adopting neural networks to fit their own research interests.

I aspired to be one of them, so during my first meeting with Tim when he asked me what my research interest is, “neural networks” was one of the keywords. While some aspects of that answer have stayed unchanged, my understanding of using neural networks for research has taken multiple turns. This definitely sounds facetious looking back, but before I trained my first network on research data, I envisioned everything to go smoothly. To be fair, all the online courses made it feel so effortless: you apply this model (which they tell you) to this highly curated data, and voila, you get an almost 100% classification accuracy. However, training a network for research purposes has almost never been so easy.

A neural network is only as good as its training data

One of the most common problems any researchers will encounter is noisy data. As a birdsong researcher, I’ve had my share of noisy data: almost all recordings of birdsongs are noisy due to the nature of sound recording, albeit some less noisy (in a controlled environment) than others (in the wild). Thanks to some of my labmates, we now have a robust denoising algorithm for sound pressure waveforms, but before that came around, it was frustrating to use noisy data to train a network only for it to fail because then one wouldn’t know whether it was because the data was noisy, or because the hypothesis was wrong. So many times, we tried to tune our models to improve its performance, only to realize the data was noisy, the collection process was unorganized, and/or the training and testing datasets were contaminated. I learned that no matter what network model I use and what parameters I tune it to, my model is only as good as the data I train it with. I have always kept this in mind since I started collecting data for my own thesis.

Neural networks are only tools to reach your conclusions

When I first started, I was a victim of the shiny new toy syndrome, where I would get distracted by the latest neural networks, spend a week to implement them, realize a different network architecture better fits my research needs, and move on to the next network. There were also times where I would get too focused on optimizing a network for better performance to realize I'm missing out on the bigger picture. In the past few years, I've learned that as a researcher, I should constantly remind myself to keep my eyes on the prize that is the research goal. There can be more than one way to reach the prize--some might work better than the ones I already know. While it's sometimes worthwhile to take an excursion and explore other possibilities, it's always more efficient to take the working path and revisit the alternate paths later when I reach the goals. All the fancy new techniques, neural networks included, are only tools to reach the goals after all.

Sometimes classic computational methods reveal more information than neural networks

As I went through a few projects in my Ph.D., I started to realize that neural networks, while excel at extracting deeply embedded information, are far less interpretable than classic computational methods including classic machine learning techniques or even as simple as computed mutual information score. With the help of neural networks, we can achieve goals that are previously unfathomable, including some of the applications in previous chapters; however, we should always try to solve the problem with classic computational methods first because they not only are more efficient, but also reveal more crucial information on how the goal is achieved. Similar to the validation approaches I took in Chapter 3, I learned to use a combination of classic computational methods and neural networks to test my hypotheses.