UNIVERSITY OF CALIFORNIA SAN DIEGO

Automatic Labeling and Representation of Birdsong for Speech Prosthesis

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Electrical Engineering
(Machine Learning and Data Science)

by

Aparna Srinivasan

Committee in charge:

Professor Vikash Gilja, Chair
Professor Nicholas Antipa
Professor Timothy Gentner

2021

The thesis of Aparna Srinivasan is approved, and is acceptable in quality and Form for publication on microfilm and electronically.

University of California San Diego

2021

Passion for your work is a little bit of discovery,

followed by a lot of development,

and then a lifetime of deepening.

— Angela Duckworth

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I have been fortunate enough to spend my Master's program as a member of the Translational Neuroengineering Lab. Vikash Gilja, it has been an absolute pleasure learning from you since ECE 209 in Winter'20. You have been an amazing mentor who has constantly motivated me to be a better researcher and to think about the big picture. Your unwavering support and patience during the pandemic have been an immense boost towards making this thesis possible.

To my incredible lab mates, it has been a delight virtually working with you. I hope we meet in person more often in the future. Pablo Tostado, thanks for processing the electrophysiological and providing your insights for this thesis. 2020 was not a typical year, but you and Daril Brown were always there to brainstorm ideas and discuss the songbird prosthesis project. To the rest of the project members, your insights have been extremely valuable. Being a highly collaborative team, I'm always in awe of how seamlessly we work together.

Anirudh and Deepthi, acknowledging everything you have done for me these past two years might be a thesis by itself! Thanks for all the life advice and the fun winter breaks that kept my sanity in check. You have been a great help in materializing this thesis. Mom and Dad, I'm ever grateful to you for letting me choose my own path from as early as I can remember and for providing a nurturing environment to learn and grow. Finally, Jaiyashri, thanks for the healthy dose of distractions that helped me destress and stay grounded during graduate school.

ABSTRACT OF THE THESIS

Automatic Labeling and Representation of Birdsong for Speech Prosthesis

by

Aparna Srinivasan

Master of Science in Electrical Engineering
(Machine Learning and Data Science)

University of California San Diego, 2021

Professor Vikash Gilja, Chair

Songbirds are widely studied as an animal model to accelerate the development of neurally driven speech prostheses. Consequently, the analyses carried out on the songbird model are dependent on high-quality labeled datasets. Building such datasets is usually laborious, as it involves manually annotating their vocal behavior, after which, neural activity can be decoded into vocalizations. However, the direct translation to vocal behavior is quite challenging since songbird vocalizations are typically recorded at high sampling rates to capture the rapid changes in behavior.

In this thesis, such problems are addressed using data-driven approaches. To reduce

the efforts involved in manual annotations, deep learning models are explored for automatic labeling of vocalizations. A model based on convolutional and recurrent layers called TweetyNet is trained with a small amount of manually labeled data comprising features from audio. It is shown to achieve high frame-level sensitivity and high temporal precision in annotating the vocalizations of adult male zebra finches whose recordings were collected in-house. Alternatively, a WaveNet-based fully convolutional model trained directly on audio, is also shown to provide high temporal precision in annotations.

To minimize the complexities involved in the direct translation of neural activity to behavior, an intermediate stage is introduced in the neural decoding pipeline. This stage will encode a low-dimensional representation of the behavior from which vocalizations can be reconstructed. A Vector Quantized Variational Autoencoder is trained to learn the latent representation of zebra finch vocalizations. Additionally, from these latent representations, novel stimuli are generated for use in psychophysical experiments.

While the stereotypical behavior of songbirds is widely studied, for practical vocal prostheses it is equally important to be able to decode the non-stereotypical behavior like calls, which can be of multiple types. The different calls in zebra finches are identified using Uniform Manifold Approximation and Projection, and the spike counts are determined from their corresponding neural activity. Gaussian classification of spike counts is shown to achieve high accuracy corroborating the mutual information existing between spike counts and call type.

These techniques and analyses are believed to provide insights for the development of songbird vocal prostheses.

Chapter 1

Introduction

1.1 Motivation

Stroke, locked-in syndrome and neurodegenerative disorders often affect speech motor control and can eventually lead to irreversible loss of ability to speak [1–4]. For such patients, the quality of life could be improved by developing assistive devices like brain computer interfaces (BCIs) that help restore spoken communication. Recent research has shown that it is possible to directly decode speech articulatory information [5–7], vowels [8, 9] and even utterances from invasive neural recordings [10–12]. However, research and development in human BCIs is difficult not only due to the restrictions involved in conducting invasive clinical studies [13–15], but also due to the complexity of speech [13, 16, 17]. Therefore, by developing and testing on simpler animal models can be supplementary to clinically translatable communication prosthesis.

Songbirds have long been of interest to study sensory-motor learning. It has been observed that they learn to sing the same way human infants learn to speak [18–21]. Consequently, the neural mechanisms of motor control involved in song production can be informative of speech production in humans. They typically present a small repertoire of vocalizations that include calls [22], introductory notes [23] and syllables (elements of song) [24]. With songs that are

simpler than human speech and relative ease of performing invasive neural recordings, songbirds act as an ideal animal model for vocal behavior decoding that can accelerate the translation of BCI-based communication technology to human clinical studies [25, 26].

1.2 Existing Work

Insights into the songbird model are gained by analyzing both the neural activity [26] and the vocal behavior [27] from high-quality datasets with labeled instances of vocalizations. These datasets typically require time-consuming manual annotations to ensure precision and quality of labels. Common strategies for automated labeling like threshold based audio segmentation is generally not robust due to background noise and fluctuations in amplitude. Similarly, template matching approaches which identify vocalizations by parsing raw continuous recordings and finding correlations with a suitable template are limited by the quality of the template and the homogeneity of the recording conditions and behavior [28]. Therefore, techniques that are designed to perform well on individual recordings can neither be scaled to large datasets nor transferred to other bird individuals and species. These limitations can be circumvented by training deep learning models with a small amount of hand-labeled data [29–31]. These models learn the relevant information in vocalizations and once trained they can be used to automatically label large datasets and generalize to other bird individuals.

At the core of BCIs for songbirds lie the decoders that translate neural activity to birdsong [25]. To alleviate the complexities in translating high-dimensional data, decoding is typically performed with the low-dimensional representations of neural and behavioral recordings [26, 32–35]. Considering behavior, from its low-dimensional latent representation, the high-dimensional data can be reconstructed for downstream applications like psychophysical and closed-loop experiments. For vocalizations or audio, dimensionality reduction techniques like autoencoders are typically employed, wherein they are trained in an unsupervised and data-driven

manner to learn the representations that are deterministic points in the latent space [36, 37]. Further, as variational autoencoders [35, 38] learn latent distributions, novel vocalizations can also be generated by sampling from them. The behavioral and neural responses to these novel vocalizations can be studied by presenting them to songbirds [39, 40]. Thus, these approaches can encode the relevant information in songbird vocalizations into its latent representations.

1.3 Proposed Work

Considering the existing work on automatic labeling and representation of birdsong, the focus of this thesis is on exploring their performance on an in-house dataset that comprises simultaneous neural and audio recordings from adult male zebra finches. The aptly named TweetyNet model is considered [29] for automatic labeling and its performance is benchmarked using sensitivity and timing errors as metrics. The benefits of intra-species transferability are also explored and experiments are carried out to determine the minimum amount of hand-labeled data required to maximize the model performance. A WaveNet [41] based automatic labeler is also proposed, that can be directly optimized on audio input thereby eliminating the need for feature engineering.

An autoencoder of the variational type known as Vector Quantized Variational Autoencoder (VQVAE) [42] is explored for the latent representations of vocalizations. The model has a vector quantization bottleneck that provides discretized latent representations for the input data. The quality of the learned representations is tested by incorporating a pretrained VQ bottleneck into the TweetyNet model and measuring its performance as a labeler. An example novel vocalization is also generated from the latent representations provided by VQVAE. These analyses are supplemented with an extensive discussion on suggested best strategies and design choices that must be considered while employing such generative models.

Finally, with a curiosity to explore non-stereotypical behavior which has not been widely

studied for vocal prostheses, the neural activity leading to the production of calls is analyzed and quantified in terms of encoded information. The different types of calls [22, 43–45] in our dataset are determined using Uniform Manifold Approximation and Projection (UMAP) [46]. Spike counts are computed from the preparatory neural activity and their causal relationship to the behavior is determined with mutual information and classification experiments.

1.4 Chapter Structure

This thesis comprises three core chapters in addition to the Introduction and Conclusion. In Chapter 2, the in-house dataset is introduced followed by an in-depth analysis on automatic labeling methods. Chapter 3 presents VQVAE for latent representation, generation of novel vocalizations and an extensive discussion on suggested strategies to design the models and experiments. Chapter 4, describes the UMAP approach for determining call types and the analyses on their preparatory neural activity.

Chapter 2

Automatic Labeling of Vocalizations

Automated labeling discussed in this thesis is achieved using supervised deep learning techniques, which can be broken into four major stages: collecting and annotating the dataset of interest, preprocessing and feature engineering, modeling/training, and testing and evaluation of results. In this chapter, the four stages are discussed with the design decisions relevant at each stage. At the testing stage, the performance of the model is evaluated against different metrics and additional experiments are carried out to assess its generalizability and efficacy in limited training data paradigms.

2.1 Dataset Description

The herein presented dataset comprises electrophysiological measurements and audio simultaneously recorded from three adult male zebra finches (z007, z017 and z020). 4-shank 16/32 site Neuronexus probes with 32-channels for z007 and 16-channels for the remaining two birds were used for recording the electrophysiological data. The probes were implanted targeting the premotor brain region HVC [47, 48]. The audio was recorded through a microphone (Earthworks M30) connected to a preamplifier (ART Tube MP). The extracellular voltage waveforms and the pre-amplified audio were amplified and digitized at 30kHz using intan RHD2000 acquisition

system, Open Ephys and a custom software [49].

The audio recordings were parsed through a custom template matching algorithm to find potential instances of vocal activity, which were then manually curated to remove false positives. In these instances of activity, the vocal events were manually segmented and labeled. A unique label was provided for each syllable, introductory note and call. There were also two labels for silence: one for the gaps between song syllables, and another for the instances of silence occurring outside of the periods of a song. Additional details regarding the recording setup and labeling can be found in [26]. For this thesis, all silences are considered to be of the same type and the annotated dataset was accessed via a custom in-house Python software *BirdSongToolbox*.

An overview of the dataset is provided in Table 2.1. A motif is a consistent stereotypical sequence of syllables interleaved with silent periods that is learned from a tutor when the zebra finches were young [50]. Beyond the syllables of the motif, the male zebra finches may also produce additional syllables they optionally insert between successive repetitions of motifs. They are called intra-motif notes. In this dataset, both z017 and z020 had one intra-motif note each. In this thesis, they have been included as another syllable in the repertoire.

Table 2.1: Description of the dataset.

Bird	**z007**	**z017**	**z020**
Repertoire	5 syllables, I, C	7 syllables, C	4 syllables, I, C
# Motif, I, C	220, 413, 578	181, -, 219	269, 426, 333
Avg. Motif duration (s)	0.6659	0.7761	0.3699
# days recorded	4	2	2
Total vocalization (s)	226.23 ($\sim$3.7 min)	166.09 ($\sim$2.7 min)	156.43 ($\sim$2.6 min)
Total recording (s)	2368.19 ($\sim$40 min)	1846.35 ($\sim$31 min)	2287.61 ($\sim$38.11 min)
# channels (electrophysiology)	32	16	16

2.2 Audio Preprocessing

Prior to analysis, audio recordings are generally filtered to remove background noise. Figure 2.1 shows an exemplar spectrogram (in dB power) from around a song sample in the

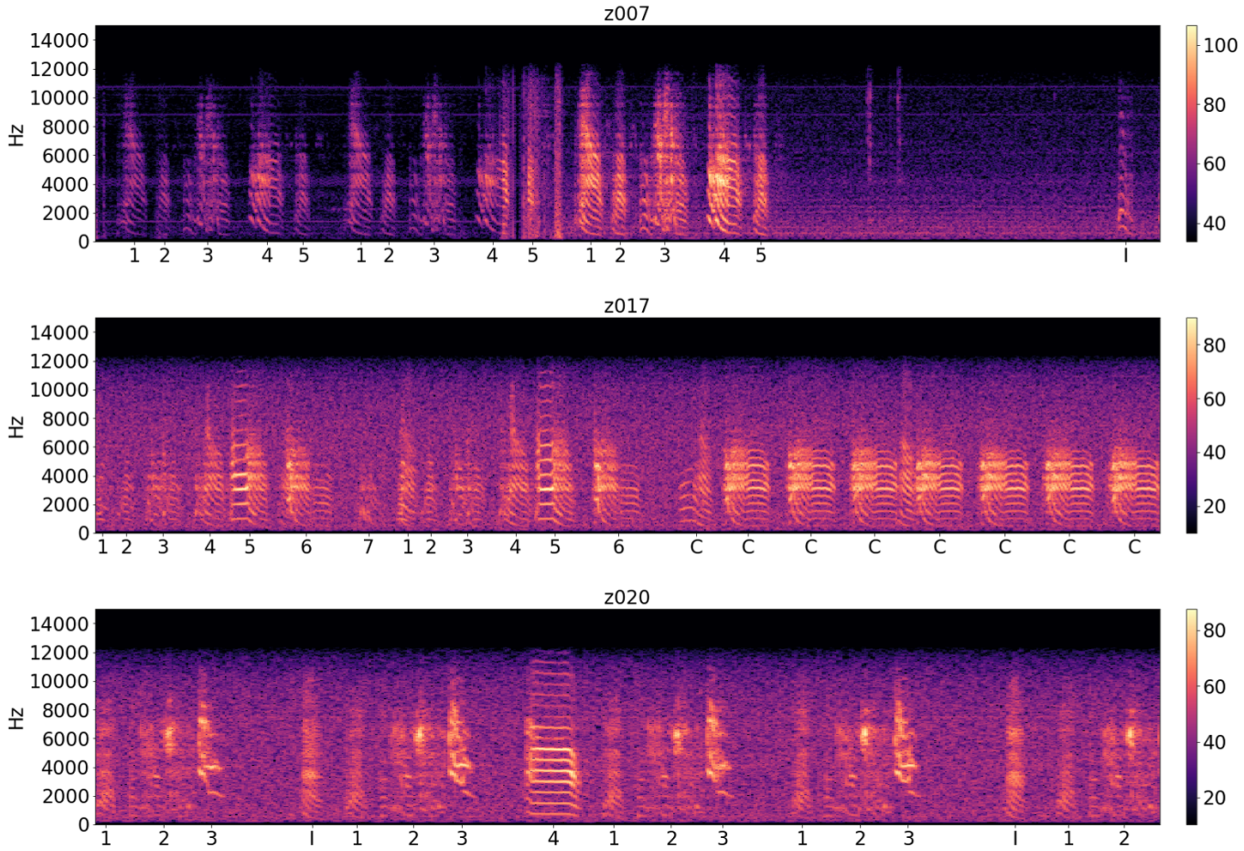

Figure 2.1: Spectrograms of audio recordings from z007, z017 and z020.

vocalization instance for the three zebra finches. The background noise in the audio recordings was reduced using a combination of thresholding and masking, based on the noise power at each frequency. It was implemented using the Python package *noisereduce* [51]. Either the noise filtered audio or the noise filtered mel spectrograms were used for training, depending on the input requirements of the deep learning models. In case of the latter, a 1024-point short-time Fourier transform (STFT) was computed on the noise filtered audio using a window size of 512, hop length of 128 and Hanning window. The spectrograms were then band-limited to 350 - 7000Hz and mapped to a mel basis with 64 bins. The timing of the vocal events in spectrograms was appropriately mapped from their labels at 30kHz sampling rate using the STFT parameters. The vocalization labels were one-hot encoded.

2.3 TweetyNet Model

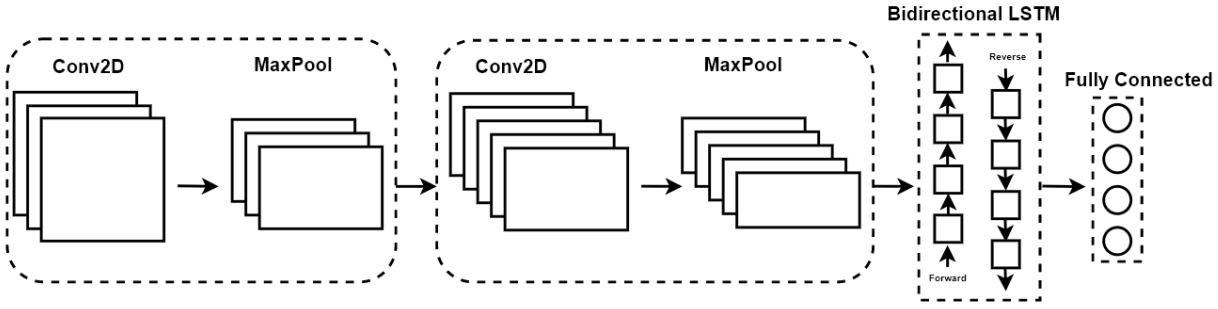

Figure 2.2: TweetyNet

Since the spectrograms of the audio recordings (Figure 2.1) reveal formant-like structures that are unique to each vocalization, a model that can capture such spatio-temporal relationships would be a good candidate for automated labeling. TweetyNet [29] is one such deep neural network that comprises two convolutional blocks [52], followed by a recurrent layer with Long Short Term Memory (LSTM) units [53] and finally a fully-connected layer as shown in Figure 2.2.

Each convolutional block has a 2D convolutional layer that captures the local spatial information in the spectrograms and a maxpooling layer that compresses the captured information with the local maximum thereby reducing the feature map size. The feature maps are padded with zeros to ensure the temporal length of the input is retained for predictions. The learnt feature maps are passed through the bidirectional LSTM layer that traverses its input in both forward and reverse directions to learn the temporal patterns in the feature maps. Finally, the fully connected layer predicts the vocalization label at each time step in the input.

2.4 WaveNet Model

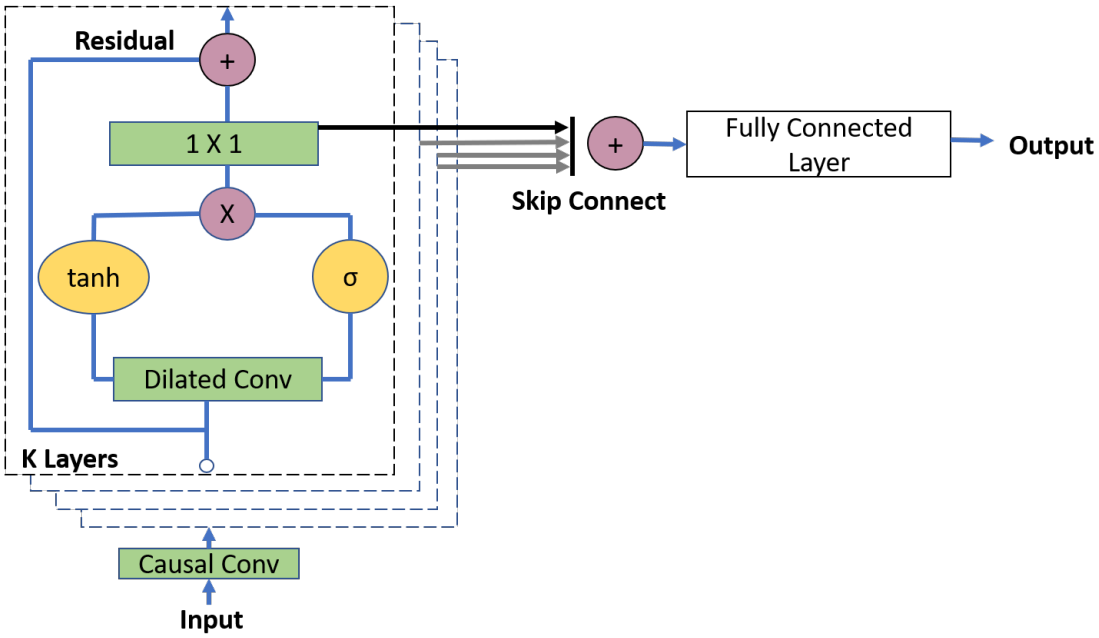

Figure 2.3: WaveNet

While the TweetyNet model requires the input to be mel spectrograms, it is possible to design models that operate directly on audio without the need for any feature engineering. Sometimes, such models could reduce the efforts involved in finding the best features for representing the information in audio. One such model is WaveNet [41], which is a deep neural

network that was specifically designed for generating raw audio waveforms, thereby making it fully probabilistic and autoregressive. However, it can also be employed as a discriminative model that returns posterior probabilities for classification. The characteristic component of WaveNet is causal convolution which ensures that the audio input is processed in sequence. Further, the receptive field of the convolutions is increased by incorporating a dilation factor which allows the filters to be applied over an area larger than their length by skipping input values with a certain step. Unlike pooling and strided convolutions, dilation helps in retaining the same length as the input.

As shown in Figure 2.3, a typical WaveNet model comprises layers of causal and dilated 1D convolutions with residual [54] and skip connections. Since they don't have recurrent connections, they are typically faster to train than recurrent neural networks especially when applied to very long sequences. The residual connections help circumvent the vanishing gradient problem [54]. At any layer k the dilation factor is 2^{k-1} and the convolutions are followed by gated activation unit [55] given by,

$$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x}\right) \odot \sigma\left(W_{g,k} * \mathbf{x}\right)$$

where * denotes a convolution operator, $\odot$ denotes an element-wise multiplication operator, σ is a sigmoid function, k is the layer index, f and g denote filter and gate, respectively, and W is a learnable convolution filter. Finally, from the sum of all skip connections, the fully connected layer predicts the vocalization label at every time step in the input.

2.5 Experimental Setup

The audio data was divided into a train and test set. The test set consisted of a vocalization instance from each day of recording for each zebra finch. The train set comprised the remaining vocalization instances. The audio (for WaveNet) or mel spectrograms (for TweetyNet) in the train set were chunked into windows of length 500ms with a stride of 50ms. Chunks with no vocal

events were discarded. The remaining chunks were randomly split into five folds for a stratified five-fold cross validation setup with 80% for training and 20% for validation. In each fold, the training, validation and test sets of data were z-score normalized. In case of mel spectrogram, the mean and standard deviation were computed along each frequency bin. Further, to control the range of values, log transformation was applied to the mel spectrograms.

After a reduced grid search, the optimal parameters for the TweetyNet model were found to be the same as reported in [29]. The convolutional layers have 32 and 64 filters respectively and filter size of 5×5. The filters were convolved with stride of 1 and padding of 2. The two maxpooling layers had pool size of 8×1, essentially pooling the feature maps along the frequency axis. Finally, the bidirectional LSTM and fully connected layer had 64 and 10 hidden units respectively. The number of classes (no. of output units in the fully connected layer) was chosen to be 10 since at most the zebra finches had a repertoire that consisted of seven syllables, introductory note and call. Further, by retaining the same architecture for all individuals, transfer learning experiments can be easily carried out to test the model's generalizability. The activations for the convolutional, LSTM and fully connected layers were Rectified Linear Units [56], tanh and softmax respectively. The net receptive field of TweetyNet (after the two convolutional blocks) is 36ms and the number of trainable parameters is 119,946.

The WaveNet model was designed to have six 1D convolutional layers with 64 filters of size 15. The first layer of the model performed a pointwise convolution with 64 filters. The convolutions were followed with gated activations as explained in Section 2.4. The sum of skip connections from the six layers was fed to a fully connected layer with 10 hidden units and softmax activation. The net receptive field of WaveNet (after the six convolutional layers) is 60ms and the number of trainable parameters is 763,786.

Both models were trained with a five-fold cross-validation setup using categorical cross-entropy loss and Adam optimizer [57] for a maximum of 50 epochs. Early stopping was invoked if the validation loss did not improve over five consecutive epochs and the model with the best

validation accuracy was saved. The average performance across folds is reported for the test set. It must be noted that the frame-level performance is computed from spectrogram frames for TweetyNet and audio frames for WaveNet.

2.6 Results and Discussion

The metrics described in this section have been derived from the confusion matrix with entries at the frame level. Sensitivity or the true positive rate for each vocalization is measured as the percentage of correct predictions out of all the frames with the vocalization.

2.6.1 Individually Trained Models

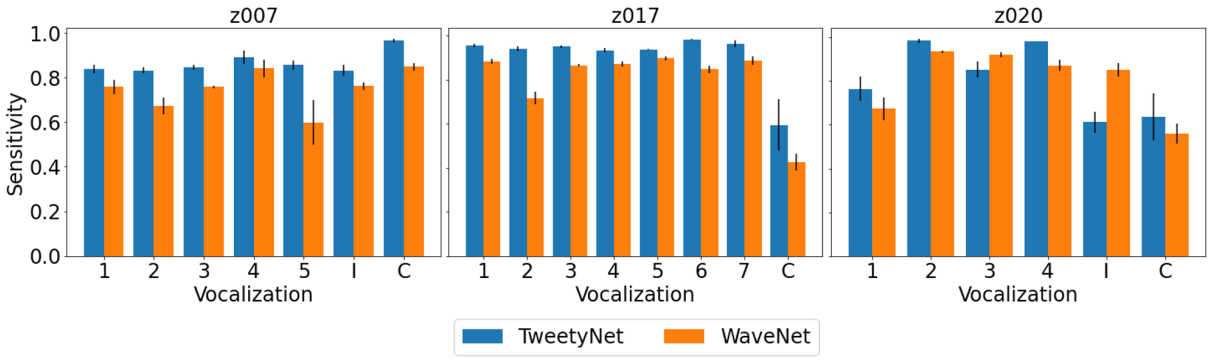

Figure 2.4: Frame-level sensitivity of TweetyNet and WaveNet across the repertoire of z007, z017 and z020.

Figure 2.4 shows the frame-level sensitivity of TweetyNet and WaveNet classification models to the vocalizations in each zebra finch. In case of TweetyNet, it is observed that the sensitivity to all vocalizations is above 0.8 for z007 and above 0.9 for all the syllables in z017. In z017, the sensitivity to call is observed to be the lowest due to the misclassification of call frames as syllable '6'. The spectrograms for call and syllable '6' as shown in Figure 2.1 had a very similar structure which led to confusion during classification as reported in the confusion matrix in Figure 2.5. For z020, it is observed that the frame-level sensitivity to vocalizations is

highly fluctuating compared to other zebra finches. Particularly for calls, about 15% of its frames were misclassified as silence and introductory note as shown in the confusion matrix in Figure 2.5. Further, this might have also been due to the significant background noise observed in z020 audio recordings as shown in Figure 2.1.

It is observed that the performance reached by the WaveNet classification model is typically lower than that of TweetyNet. It must be noted that the two models are not comparable since they operate on different inputs and the sensitivity has been computed on time frames sampled at different rates. However, the performance trend across vocalizations is found to be similar between WaveNet and TweetyNet. In case of z007 it is observed that the frame-level sensitivity to all syllables except '5' is above 0.7. The confusion matrix shown in Figure 2.6 shows that this is because 16% and 13% of syllable '5' frames were misclassified as syllable '2' and call respectively. In case of z017, the sensitivity to call is observed to be lower than that for syllables due to the same reasons outlined for the TweetyNet model. Similarly, due to poor audio quality, the frame-level sensitivity fluctuates across vocalizations of z020.

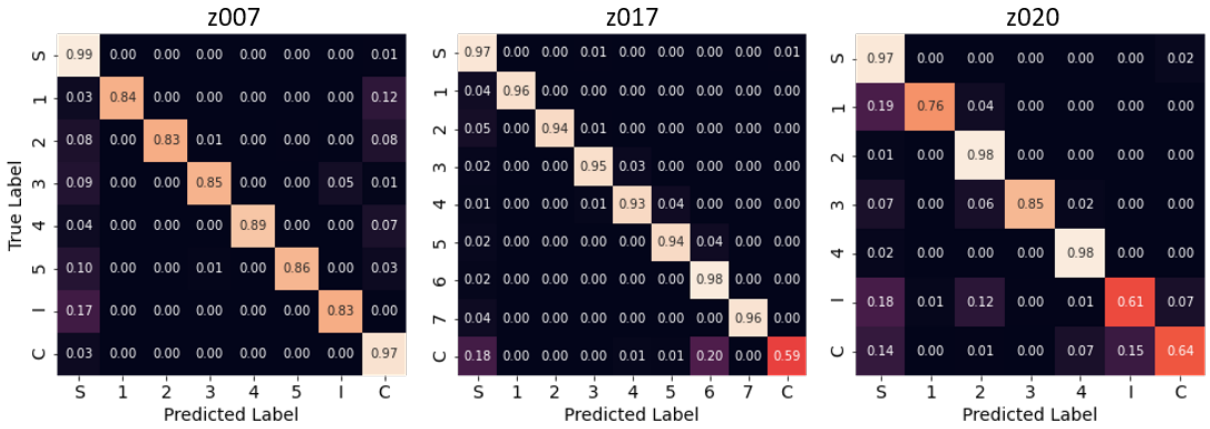

Figure 2.5: Normalized confusion matrices from TweetyNet for z007, z017 and z020.

Figure 2.5 and 2.6 show the normalized confusion matrices for the TweetyNet and WaveNet models trained individually on each zebra finch. The confusion matrices are obtained by dividing each element of the matrix with the sum of elements in the corresponding row i.e.,

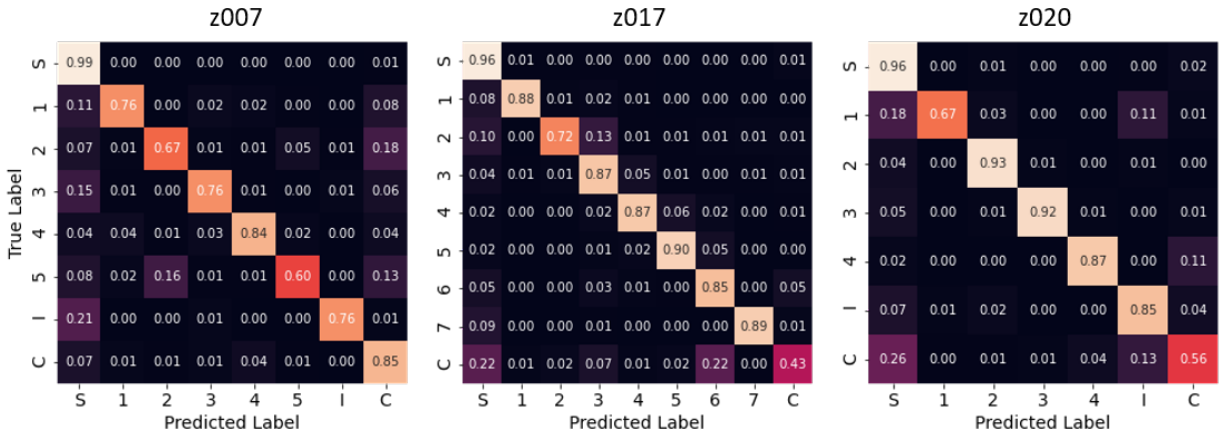

Figure 2.6: Normalized confusion matrices from WaveNet for z007, z017 and z020.

the number of true labels. Thus, the diagonal elements represent the sensitivity or the percentage of frames that were correctly classified for each vocalization. It is observed that the confusion matrices show similar values for both TweetyNet and WaveNet across all zebra finches, indicating that the performance is driven by the same underlying property in the acoustics. Thus, a common discussion is provided here that is applicable to both models.

In z007, it is observed that about 8-17% of the syllable frames have been misclassified as either silence or call. A majority of such misclassifications were found to occur in one of the test vocalization instances of z007 which had varying audio quality. The test vocalization instance had a period of low amplitude and low noise (Figure 2.1) wherein the syllables were predominantly classified as silence or call. This indicates that both TweetyNet and WaveNet models are not robust to drastic changes in amplitude while recording. In case of z017, it is observed that the sensitivity to call is the least at 0.59 for TweetyNet and at 0.43 for WaveNet. About 20% of the call frames have been misclassified as syllable '6'. This stems from the similarity between 'long call' and syllable '6' which can be noticed from the spectrogram shown in Figure 2.1. In the case of z020 it is observed that a typical reason for the drop in sensitivity is due to the misclassification of vocalizations as silence. Additionally about 13% and 15% of call frames have also been misclassified as introductory note by WaveNet and TweetyNet respectively. In general, the high

sensitivities reveal that both models not only learn the patterns in the vocalization elements but also the background noise in the recordings. Table 2.2 summarizes their overall frame-level sensitivity i.e the average frame-level sensitivity across vocalizations.

Table 2.2: Overall frame-level sensitivity of TweetyNet and WaveNet for z007, z017 and z020.

Overall Sensitivity	z007	z017	z020
TweetyNet	0.86	0.90	0.80
WaveNet	0.74	0.79	0.80

Table 2.3: Onset, offset and transient timing errors of TweetyNet and WaveNet for z007.

z007		1	2	3	4	5	I	C	Overall
TweetyNet	Onset (ms)	7.24	7.90	23.29	11.66	5.96	5.87	3.03	9.28
WaveNet		7.97	**5.12**	**20.71**	**3.58**	**5.01**	**4.96**	**2.21**	**7.08**
TweetyNet	Offset (ms)	9.67	9.67	23.90	9.50	7.74	7.77	3.99	10.32
WaveNet		**5.86**	**7.65**	**8.35**	**4.10**	8.71	**5.95**	**3.39**	**6.29**
TweetyNet	Transient (ms)	0	0	1.1	0.47	0.05	0.05	0	0.23
WaveNet		6.03	4.61	11.41	10.79	9.76	3.73	8.16	7.78

Table 2.4: Onset, offset and transient timing errors of TweetyNet and WaveNet for z017.

z017		1	2	3	4	5	6	7	C	Overall
TweetyNet	Onset (ms)	1.72	4.41	5.21	2.78	1.63	3.27	4.84	24.31	6.02
WaveNet		2.11	**3.38**	**1.88**	**2**	**1.164**	**2.24**	**3.16**	**7.53**	**2.93**
TweetyNet	Offset (ms)	2.98	3.43	3.81	4.46	6.34	2.89	6.35	23.25	6.69
WaveNet		3.61	4.11	5.95	**1.39**	**4.31**	2.94	**2.92**	**6.44**	**3.96**
TweetyNet	Transient (ms)	0	0	0	0.25	0	0.17	0	2	0.30
WaveNet		1.54	8.99	8.64	7.10	4.74	18.16	4.27	34.29	10.97

Table 2.5: Onset, offset and transient timing errors of TweetyNet and WaveNet for z020.

z020		1	2	3	4	I	C	Overall
TweetyNet	Onset (ms)	5.30	1.56	1.47	0.46	9.14	17.68	5.93
WaveNet		6.98	2.49	**0.59**	0.83	**1.72**	**12.56**	**4.2**
TweetyNet	Offset (ms)	7.57	2.65	5.54	1.8	9.32	18.22	7.52
WaveNet		**5.43**	**1.49**	**2.7**	**0.85**	**2.32**	**5.23**	**3**
TweetyNet	Transient (ms)	0	0.16	0	0	0	1.68	0.30
WaveNet		2.65	4.12	1.93	12.38	2.24	15.30	6.44

While sensitivity provides the percentage of frames correctly classified for each vocalization, they do not reveal the temporal precision associated with detecting the onset and offset of vocal events. Temporal precision in labeling is of crucial importance particularly in applications like neural prosthetics that rely on neural decoding. When studying the causal relationship between neural activity and output behavior, inaccuracies in determining the onset and offset times of the behavior of interest lead to the observation of neural activity that may be unrelated. This is particularly true when analyzing fast occurring behavior, like birdsong. Table 2.3, 2.4 and 2.5 detail the timing errors made by TweetyNet and WaveNet at detecting the onset and offset of vocalizations along with any transient errors in classification during the vocalization. For onset and offset errors, it is the number of frames that were misclassified before and after the actual onset or offset of a vocalization respectively. For transient errors, it corresponds to the number of frames misclassified within the vocalization. The number of frames was divided by the sampling rate in the case of WaveNet or multiplied by 4ms ($\approx$ spectrogram hop length / sampling rate) for TweetyNet. The overall timing error is computed as the average error across all vocalizations. From the tables it is observed that overall timing errors do not exceed 11ms for both TweetyNet and WaveNet. Again, it must be noted that the two models cannot be compared as they operate on different inputs. WaveNet has a lower (in bold) onset and offset errors than TweetyNet whereas it is the opposite in case of transient timing errors. In fact, transient errors in the TweetyNet model approach 0ms. Of all the timing errors presented, it is critical to minimize the onset and the offset errors to ensure that the correct period of neural activity is decoded. By supplementing with an ad-hoc post-processing stage based on the typical duration of vocalizations and silence, the transient errors can be further minimized.

Between the two models explored here, TweetyNet is one of the best models for automatic labeling. With roughly 120,000 learnable parameters, it trains faster and is not compute intensive as the model is trained on mel spectrograms. On the other hand, the WaveNet takes longer to train as it is a larger model that is applied to audio which is sampled at 30kHz. However, if

WaveNet was to be compared to an equivalent deep learning model that had recurrent layers for processing audio, then it is highly likely that it uses less resources as it consists of only convolutional layers. Further, it eliminates the need for feature engineering which can introduce an additional overhead in the labeling pipeline. Its performance can be boosted with more training data (augmentation) and it can also be leveraged as a generative model. Given the current limitations of the in-house dataset and based on the above analyses, additional experiments are performed only on TweetyNet.

2.6.2 Transfer Learning

Since the zebra finches have a limited repertoire and a highly stereotypical song, there exist similarities between the vocalizations of different individuals. These similarities can be exploited to reduce the training time and the amount of training data required for automatically labeling a new individual. This hypothesis is tested by carrying out a transfer learning experiment, wherein the TweetyNet model trained on one individual is tested on a different individual. Subsequently, the last fully-connected layer is randomly initialized and selectively retrained, while the remaining layers are not updated with the gradients.

Table 2.6: Overall frame-level sensitivity of TweetyNet, pretrained on one individual and transferred to another individual with selective retraining.

Overall Sensitivity		Selectively Retrained		
		z007	z017	z020
Pretrained	z007	**0.86**	0.72	0.57
	z017	0.75	**0.90**	0.65
	z020	0.74	0.75	**0.80**

Table 2.6 shows the overall frame-level sensitivity of the TweetyNet model that has been pretrained on one individual followed by the last layer being selectively retrained on another or same individual. It is observed that the overall sensitivity is the highest when the model is retrained on the same individual and the performance is found to match that of the fully trained

model reported in Table 2.2. Further, these models were found to achieve convergence much faster during training. An overall sensitivity of 0.75 is achieved when the models are transferred between z007 and z017. However, in the case of z020 it is observed that the transferred model performs significantly lower than a model retrained on z020. This is because the audio recordings of z007 and z017 are relatively less noisy than that of z020 and the dissimilarity in the vocal behavior as shown in Figure 2.1. Consequently, the transferred model does not perform well under mismatched noise conditions. Further, it is interesting to note that even though z020 has the smallest repertoire, TweetyNet model transferred from z020 performs on par with the models transferred between z007 and z017, indicating similarities in the vocalizations.

The benefits of transfer learning could be further extended by using models that have been trained on multiple individuals. These could be followed with full retraining to achieve faster convergence than a randomly initialized model.

2.6.3 Effect of Training Data Duration

As the TweetyNet model is a data-driven supervised learning technique, its performance is expected to increase with more training data until it plateaus to a maximum reachable performance after which additional training data has less/no impact on the performance. Therefore, the minimum amount of data required for maximum performance can be determined by gradually incrementing the duration of vocalizations in the training data. This could further reduce the efforts towards manual labeling.

For this analysis, the vocalization instances were randomly selected and grouped until the total vocalization duration equaled t seconds. If the total duration extended beyond the requirement, then it was clipped at the target duration such that all syllables were still present in the training set. For each individual, 5 such distinct groups were created for $t \in \{2, 5, 8, 10, 20, 30, 40, 50, 60, 70, 80, 90\}$ seconds. The groups of vocalization instances were chunked accordingly and the models were trained (without cross-validation) as mentioned in

Section 2.5. The average test set performance across the five models (one per group) is reported.

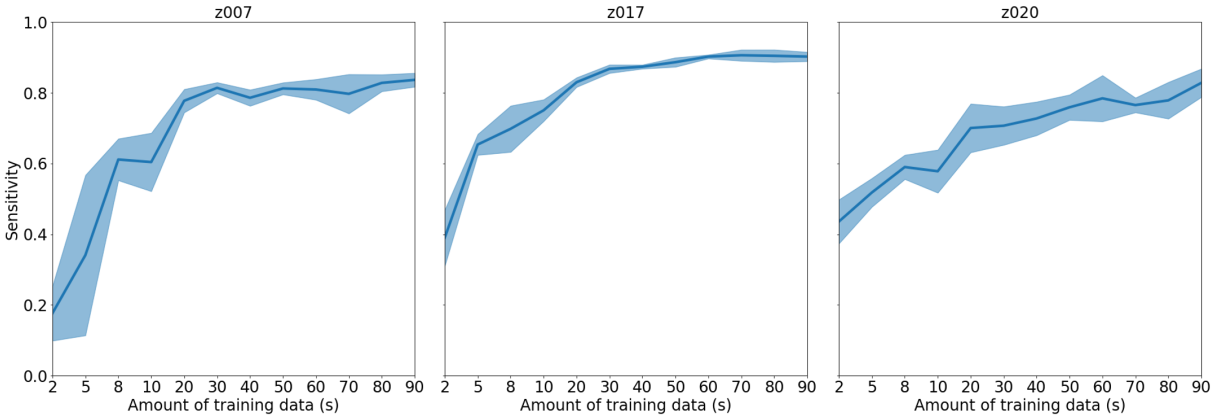

Figure 2.7: Overall frame-level sensitivity of TweetyNet with varying duration of training data.

Figure 2.7 shows the overall frame-level sensitivity of TweetyNet trained with varying duration of training data. It is observed that the sensitivity saturates around 30s for z007, 60s for z017 and 90s for z020. The need for more training data in case of z017 and z020 is anticipated as their audio recordings are of much lower SNR than that of z007 as shown in Figure 2.1. Though definitive conclusions cannot be drawn regarding the exact amount of training data that would be required to automatically label the vocalizations of a new individual, this analysis provides an insight that it is not necessary to manually annotate the entire dataset ($>$ 2.5 minutes of vocalizations). Coupled with transfer learning, the estimated amount of annotated data could be further reduced.

In conclusion, the experiments and analyses on the TweetyNet reveal that it is a highly sensitive model with low timing errors and that it can achieve high performance with small amounts of hand-labeled data. It is hypothesized that the performance could be further improved with data augmentation and transfer learning. While the WaveNet model was not explored in such detail, it could be the optimal labeling strategy as it eliminates the need for audio feature engineering and is bound to perform well under data-rich conditions.

Chapter 3

Latent Representation of Vocalizations

Latent or low-dimensional representation of audio (mel spectrogram) can be realized with dimensionality reduction techniques like UMAP [46] and t-SNE [58], or deep learning approaches like autoencoders [36] and variational autoencoders (VAEs) [59]. In this chapter, a particular type of variational autoencoder is explored, that retains its generative power while discretizing the latent space. The capabilities of the model are tested in conjunction with the automated labeling strategy discussed in Chapter 2. Additionally, its comparison to typical VAEs and the design choices for representation are addressed in the discussion.

3.1 Vector Quantized Variational Autoencoder

Vector Quantized Variational Autoencoder (VQVAE) [42] is a type of generative model that combines VAE with discrete latent representations via Vector Quantization (VQ). For the discrete latent random variables z given the input data x, the posterior ($q(z|x)$) and the prior ($p(z)$) distributions are categorical and the samples drawn from these distributions correspond to the indices of an embedding table. These embeddings (e_j for $j \in 1,2,...,K$) are then used as an input to the decoder. The posterior categorical distribution probabilities are defined as one-hot encodings as follows:

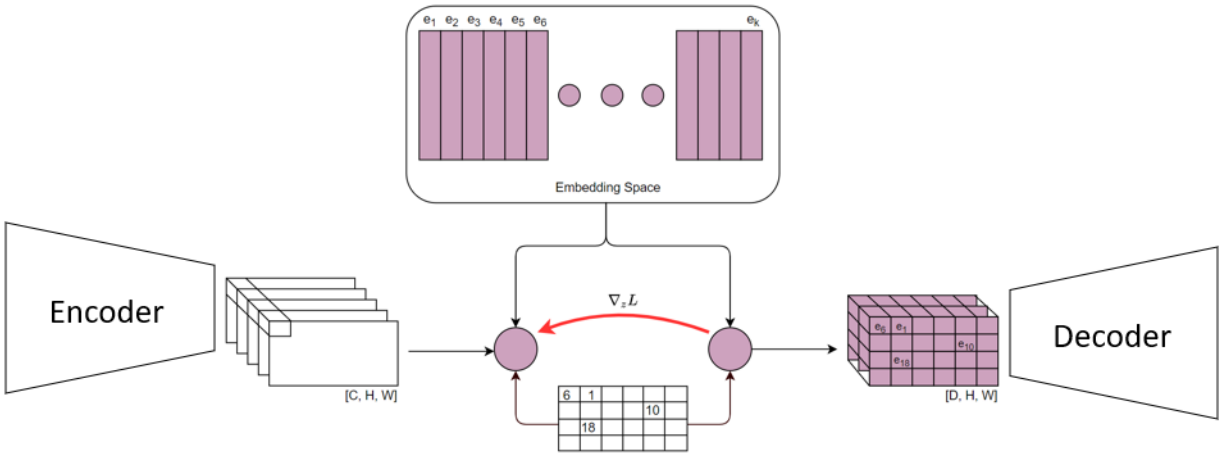

Figure 3.1: Vector Quantized Variational Autoencoder

$$q(z=k|x) = \begin{cases} 1 & \text{for k} = \text{argmin}_j \left\| z_e(x) - e_j \right\|_2 \\ 0 & \text{otherwise} \end{cases}$$

where $z_e(x)$ is the output of the encoder network. The distribution $q(z=k|x)$ is deterministic, and by defining a simple uniform prior over z, the KL divergence between them becomes $\log K$. The representation $z_e(x)$ is then passed through the discretization bottleneck followed by mapping onto the nearest element of embedding as follows:

$$z_q(x) = e_k, \quad where \quad k = argmin_j \left\| z_e(x) - e_j \right\|_2$$

As shown in Figure 3.1, each element in the encoder output (size $C \times H \times W$) is replaced with the nearest of the K embeddings (of D dimensions; $D << C$) to generate a sample of size $D \times H \times W$ for the decoder. Since there is no gradient defined for the above equation, the gradient is approximated using the straight-through estimator [60] wherein the gradients are copied from the decoder input to the encoder output. The loss function of the VQVAE model is formulated as,

$$L = \log p\left(x \mid z_q(x)\right) + \| \operatorname{sg}[z_e(x)] - e\|_2^2 + \beta \|z_e(x) - \operatorname{sg}[e]\|_2^2$$

where,

- $\log p\left(x \mid z_q(x)\right)$ is the reconstruction error measured as the mean squared error between the input and the output in terms of variance.
- $\| \operatorname{sg}[z_e(x)] - e\|_2^2$ is the codebook loss which is used for learning the embedding space e and updating the embedding vectors.
- $\|z_e(x) - \operatorname{sg}[e]\|_2^2$ is the commitment loss that helps control the volume of the embedding space and ensures that the encoder commits to an embedding.

β is the commitment cost and sg stands for the stopgradient operator. When the model is trained, the stopgradient operator acts as the identity function in the forward direction, but stops the accumulated gradient from flowing through that operator in the backward direction, thus effectively constraining its operand to be a constant that doesn't get updated. The decoder optimizes the reconstruction loss, the encoder optimizes the reconstruction and the commitment loss and the embeddings are optimized by the codebook loss. Thus, the VQVAE can make effective use of the latent space by successfully modeling relevant features that usually span multiple dimensions in the data space as opposed to focusing on noise or imperceptible details.

3.2 VQVAE vs. VAE

In a typical VAE framework, the posterior and the prior are assumed to be continuous normal distributions with diagonal covariance, which allows for the Gaussian reparameterization trick to be used [59]. Further, the prior is unit Gaussian and is retained as a constant. Consequently, at certain times VAEs can suffer from large variance. Whereas, in VQVAE, the latent space is

discretized and assumed to have uniform prior. However, the prior is not static but learnt via backpropagation. The model relies on vector quantization, which makes it simple to train and prevents it from experiencing large variance.

The VAE loss is the combination of the reconstruction error and the KL divergence between the learnt Gaussian distribution and prior (unit Gaussian). For VQVAE, since the prior is uniform and the posterior is categorical the KL divergence is essentially a constant and equals log K, where K is the size of the embedding space. Apart from the reconstruction error, the VQVAE loss has 2 additional terms called codebook loss and commitment loss which help in updating and controlling the size of the embedding space.

VAEs can sometimes experience "posterior collapse" especially when coupled with a powerful autoregressive decoder. In such situations, the latents are often ignored, i.e. the "learnt" latents become independent of the input and are constants and the decoder being extremely powerful, tries to reconstruct the input by ignoring the latents. On the other hand, "posterior collapse" does not arise in VQVAE since the nearest embedding to the encoder input is always fetched and propagated to the decoder input, and is subsequently updated by backpropagating the VQVAE loss.

Though VQVAE appears to be a better model for latent representation than a typical VAE, the performance of both models depends on the dataset (size and quality), encoder-decoder architecture and the dimensionality of the latent space. The VQVAE inherently introduces discontinuities at the decoder input due to the discretization of the latent space. These discontinuities can propagate and lead to poor reconstructions. Similarly, both the models can learn the background noise as well, since they try to minimize the reconstruction error. Thus, choosing an optimum compression ratio (= input dimension / latent space dimension) is critical to optimize their performance.

3.3 Experimental Setup

A two-stage approach was followed to test the efficacy of VQVAE in learning discrete latent representations for zebra finch vocalizations. The two stages were,

- Stage 1: An autoencoder of vanilla or VQVAE type is trained to minimize the reconstruction error or the loss described in Section 3.1 respectively. The encoder consists of the two convolutional layers of TweetyNet (without maxpooling) and the decoder was designed as an inverted encoder with two transpose convolutional layers.

- Stage 2: The decoder of the autoencoder is replaced with TweetyNet's decoder that comprises the bidirectional LSTM and fully connected layer. The decoder is selectively trained to optimize the categorical cross-entropy loss. The weights of the pretrained encoder and the embeddings in case of VQVAE are kept frozen.

For VQVAE the size and the dimensions of the embedding space were chosen as 32. The parameters of the layers from the TweetyNet model were retained as mentioned in Section 2.5. Due to memory constraints that stemmed from a high dimensional latent Gaussian representation, a TweetyNet based VAE was not realizable and hence is not included in the analyses. The TweetyNet classification model with the VQ bottleneck is called 'TweetyNet-VQVAE'. Similarly, the TweetyNet classification model pretrained as a vanilla AE is termed 'TweetyNet-AE'. During Stage 1, all the models were trained as mentioned in Section 2.5 with the same 5-fold cross-validation setup. At Stage 2, the training is dependent on the type of analysis and is explained in the corresponding Section. For comparison, a TweetyNet model without the maxpooling layers is fully trained and termed as 'TweetyNet (no pooling)'.

3.4 Results and Discussion

3.4.1 Effect of Amount of Training Data Duration

As outlined in Section 2.6.3, the same experiment is carried out with the new models namely, TweetyNet-VQVAE, TweetyNet-AE and TweetyNet (no pooling). In case of the first two models, the two-stage approach is followed as explained in Section 3.3. During Stage 2, when the decoder is selectively trained, the duration of vocalizations in the training set is varied. The average test set performance across the five models (one per group) is reported.

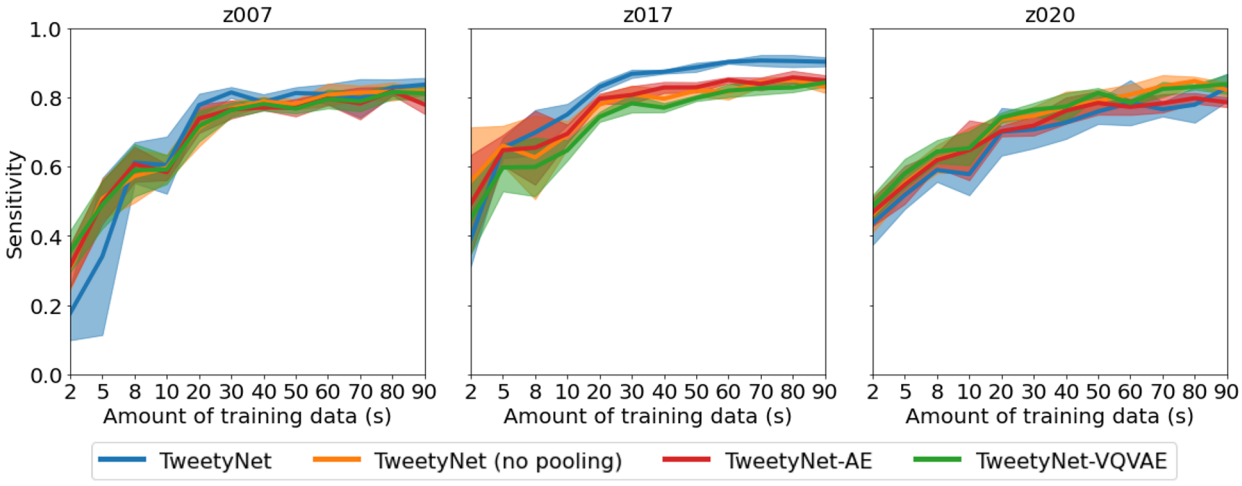

Figure 3.2: Overall sensitivity of TweetyNet, TweetyNet (no pooling), TweetyNet-AE and TweetyNet-VQVAE with varying duration of training data.

Figure 3.2 shows the overall frame-level sensitivity for different duration of training data. Generally, across all zebra finches, it is observed that the above-mentioned models have performances that are very close to the original TweetyNet as well. The similar trend in performance indicates that the embedding space in 'TweetyNet-VQVAE' has effectively learnt an embedding space encoding the information in the mel spectrograms. Given the input has 64 mel bins and 118 (= 500ms at 30kHz sampling rate) time steps and the discrete latent space is 32 dimensional, the VQVAE model learns a compressed representation that is 236 times smaller than the input. The compression ratio is measured as the ratio of the input dimension to the latent space dimension.

Comparing them with the original TweetyNet that had maxpooling layers ("TweetyNet"), it is observed that they have a higher overall sensitivity than the latter when the duration of training data is less than 20s for z007 and z020 and less than 5s for z017. The absence of pooling preserves the high dimensional feature maps benefitting the model when trained on limited data. Overall, the performance of the vanilla AE and the VQVAE are similar. While the AE model learns the mapping function that provides the best reconstruction from the latent representation, the VQVAE model learns discrete latent embeddings from which new vocalizations can be generated.

3.4.2 Vocalization Generation

The most powerful aspect of VQVAE is its generative capabilities that can be leveraged to design new stimuli for psychophysical or closed-loop experiments. To analyze the quality of the generated vocalizations, the discrete latent representations z_q from TweetyNet-VQVAE are obtained for two calls – short and long – from z007's mel spectrograms. With w and $1-w$ as the weights corresponding to latent representations for z_{q_short} and z_{q_long} respectively, novel vocalizations are generated by reconstructing the weighted latent representation z_{q_new}. Here w is varied from 0 to 1 in steps of 0.1 and the generated vocalizations are shown in Figure 3.3.

It is observed that as the weight associated with the latent representation of the long call is increased the generated mel spectrogram also morphs into appearing more like a long call. Introducing such variations in vocalizations can be of use in psychophysical experiments, wherein the discriminability between vocalizations and their perception is typically measured by monitoring the behavior and/or neural activity of participants. The reconstruction error between the weighted input mel spectrograms and the generated vocalizations was found to be ≈ 0.02 in terms of explained variance. Finally, the options are endless in terms of the weighting schemes and the combination of input vocalizations to design new stimuli, which when created and played in a sequence can be learnt by a young male zebra finch.

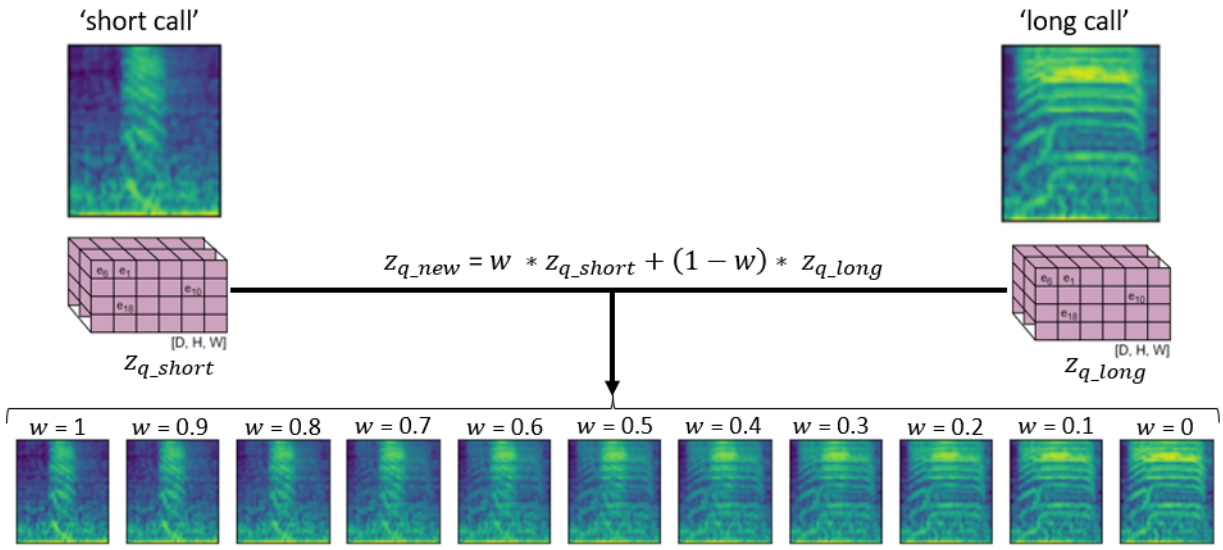

Figure 3.3: An example vocalization generated from the weighted latent representation of a long and a short call during the first stage of TweetyNet-VQVAE.

3.4.3 Audio Features for Latent Representation and Automatic Labeling

While the reconstruction errors of the TweetyNet-VQVAE model were low (≈ 0.02), when the reconstructed mel spectrograms of the test set were inverted to audio using the Griffin-Lim reconstruction algorithm [61], the resultant audio had different prosody and sounded extremely noisy with amplified high-frequency harmonics. These harmonics could be attributed to reverberations within the semi-anechoic chamber in which the audio was recorded. The poor reconstruction of audio indicates that the embedding space of VQVAE not only encodes the information in the vocalizations but also the background noise. This can be circumvented by training the VQVAE such that the embedding space does not fit the background noise. It must also be noted that the current VQVAE approach leverages the TweetyNet architecture, which need not be the best encoder-decoder architecture. Also, the reconstruction error in the loss function doesn't include any metric that relates to the quality of the reconstructed audio. Thus, there is scope for improvement in terms of the choice of encoder-decoder models, the dimensionality of the embedding space, the constraints that can be levied to prevent the embeddings from learning noise and by

incorporating perceptual loss [40, 62, 63] while training. The discrete embeddings need to be analyzed further to understand the discontinuity that they introduce in the latent representations.

On the other hand, as the use case of latent representation is to generate novel vocalizations for psychophysical experiments, it brings into question whether mel spectrogram is the best representation of birdsong. Since the mel frequency scale is based on the critical bands in the human auditory system [64], it inherently introduces bias into the model. Further, as the (mel) spectrograms are computed using STFT which involves windowing and striding, they cannot capture the rapid changes within vocalizations. However, they are good features for automatic labeling as they capture the patterns across both frequency and time which are visually discernible between vocalizations and can be easily recognized by a deep learning model. Thus, the bias established by feature engineering needs to be eliminated for the application of latent representation and audio generation. One way of achieving this is by employing an end-to-end WaveNet based architecture with a learnable latent distribution that directly encodes the useful information in the audio.

Chapter 4

Neural Analysis of Calls

Zebra finch calls can be broadly classified into two categories namely long and short, among many others [65]. Apart from the difference in duration, a long call is a type of vocalization that is learnt by young males (and not by females) while a variety of short calls are used by both males and females at a later stage [22, 45]. In the in-house dataset, all call events have been provided with the same label 'C' irrespective of their type. As shown in Figure 3.3, there exist at least two types of calls in z007. In this chapter, the types of calls in our dataset are determined with a manifold-based dimensionality reduction technique. The categories of calls thus obtained are analyzed along with the information encoded in their neural activity.

4.1 Data Preprocessing

The mel spectrograms of the audio recordings were computed as mentioned in Section 2.5 and were segmented to just the calls, which were then log scaled and padded (equally appended and prepended) with zeros to ensure all calls were of the same length.

The electrophysiological data was preprocessed as follows,

- Each electrode channel was filtered to 250 - 8000Hz with a 4^{th} order Butterworth bandpass

filter.

- For each channel the root mean square (RMS) of the signal was computed and at a threshold of 3.5 $\times$ RMS, the signal was binarized to yield a spike train. To account for the refractory period, any spike within 1ms from the previous one was not taken into consideration.

Considering up to 100ms before the onset of calls, the spike train from each channel was summed every 10ms to get the spike counts, yielding a $N \times 10$ dimensional data with N being the number of electrodes.

4.2 Uniform Manifold Approximation and Projection

The Uniform Manifold Approximation and Projection (UMAP) [46] technique constructs a high dimensional graph representation of the data and then optimizes a low-dimensional graph to be as structurally similar as possible. To construct the initial high-dimensional graph, UMAP builds a weighted graph with edge weights representing the likelihood that two points are connected. The points are connected by extending a radius from each point and connecting the points with overlapping radii. The radius is chosen locally based on the distance between each point's n^{th} nearest neighbor. As the radius grows, the graph gets built with decreasing weights for connections. Finally, by stipulating that each point must be connected to at least its closest neighbor, UMAP ensures that both local and global structures are preserved. Once the high-dimensional graph is constructed, UMAP optimizes the layout in a low-dimensional space to be as similar as possible.

In this thesis, UMAP was applied to the zero padded mel spectrograms of calls as described in the previous section with the following parameter settings – 60 for the number of neighbors, 0.25 for minimum distance and a random seed for reproducibility.

4.3 Results and Discussion

4.3.1 Types of Calls from UMAP

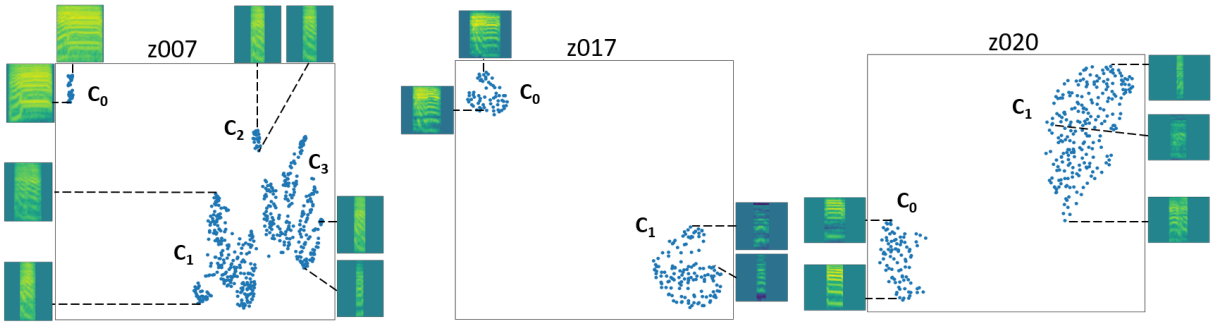

Figure 4.1: 2D projections of calls from z007, z017 and z020 using UMAP along with exemplary spectrograms for a points in the projections. Each cluster of calls is given an unique label C_i.

Figure 4.1 shows the 2D representation of the calls projected by UMAP. The mel spectrograms of a few points are displayed around the plots. For z007, it is observed that four distinct clusters are formed with the cluster C_0 corresponding to that of long calls. Considering the remaining three clusters (C_1, C_2, C_3, it is observed that they are generally shorter in duration than the calls in C_0. Hence, they can be considered short calls. Further, there exists a lot of variability within the calls in terms of their duration and the slope of the format-like structures in their mel spectrograms. In case of z017 and z020, the two distinct clusters are observed to be corresponding to long and short calls respectively. Thus, calls have been clustered into different types, primarily long and short, in the 2D space by UMAP. Subsequently, the same process was repeated for the above classified short calls. Even after a selective search on UMAP parameters, no additional clustering within short calls was observed.

4.3.2 Spike Counts and Mutual Information

For each type of call as obtained with UMAP, the average spike count computed every 10ms, up to 100ms before the onset of the vocalization is shown in Figure 4.2 for z007 and Figure

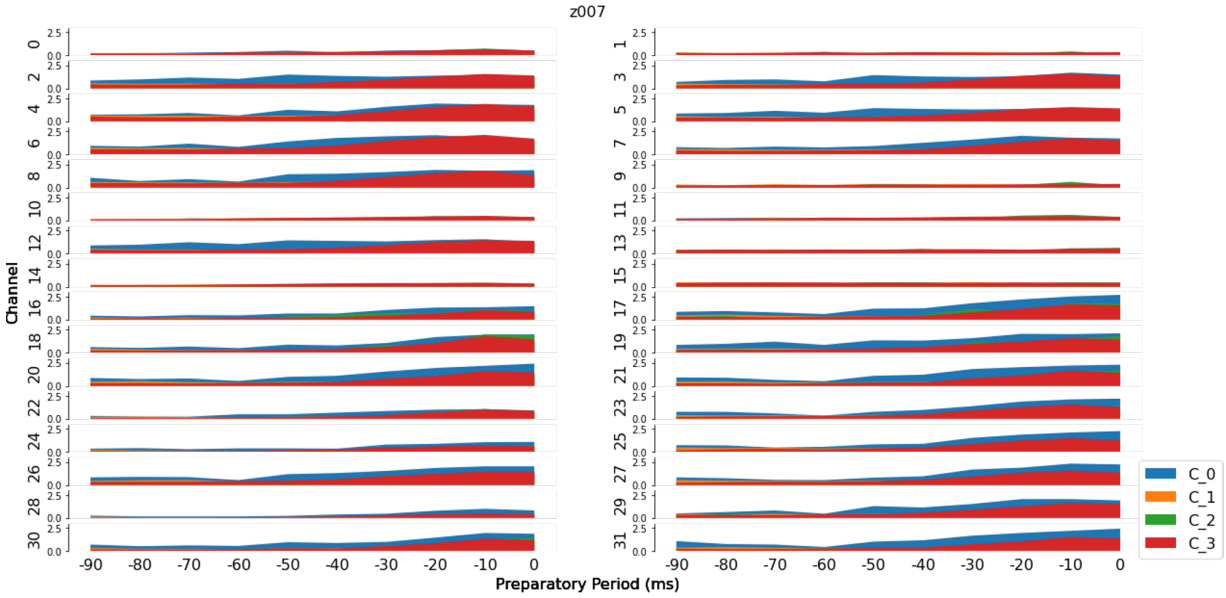

Figure 4.2: Spike counts across channels binned every 10ms, upto 100ms before the onset of calls in z007.

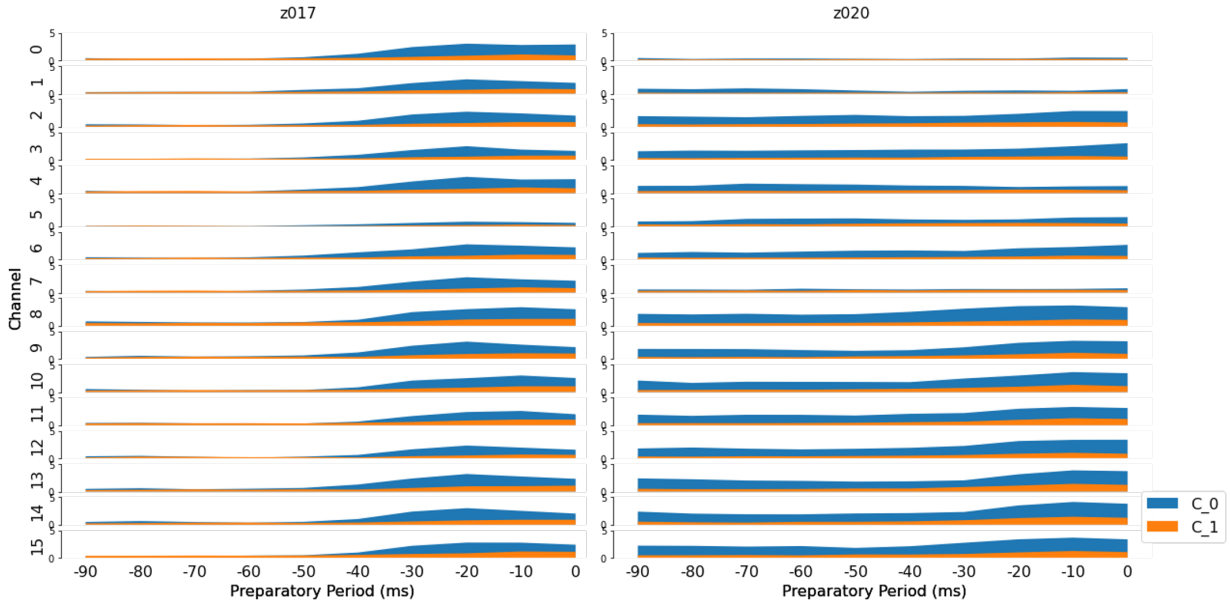

Figure 4.3: Spike counts across channels binned every 10ms, upto 100ms before the onset of calls in z017 and z020.

4.3 for z017 and z020. It is observed that the average spike count typically between 0 to 5 with the ones corresponding to long calls being generally higher than that for short calls. Not all the recorded channels show spiking activity. Further, the typical preparatory period is found to be between [-20,0]ms.

Similarly, the mutual information between the spike counts of each channel and the type of call is computed as follows,

$$I(N;L) = H(N) - H(N|L)$$

where, N refers to the spike counts or neural activity, L refers to the label, $H(N)$ is the entropy of the spike counts and $H(N|L)$ is the conditional entropy of spike counts given the call labels. The normalized mutual information (NMI) is measured as the mutual information $I(N;L)$ weighted by the mean of $H(N)$ and $H(L)$ (entropy of labels of calls). The above metrics were computed via a 2D histogram approach. Since the maximum spike count in a 10ms bin was typically under 5, the histogram bin size for the spike counts was chosen as 5 across all zebra finches.

Figure 4.4 and 4.5 show the entropy of spike counts $H(neural)$, and the conditional entropy of spike counts given label $H(neural|label)$, for every 10ms bin in the preparatory period across all channels for the three zebra finches. First, it is noticed that the channels that have high entropies correspond to the channels that also had high spiking activity, indicating that the firing activity is of high variance. However, the mutual information between the spike counts and labels is typically low as denoted by the non-overlapping regions between the two entropies. The minimum and maximum NMI across all channels is shown in Figure 4.6. Overall, the NMI between spike counts and call labels are found to be above 0 for z007 and around 0.25 for z017 and z020, indicating that there is a mutual dependence between them and that by observing one of them, some information can be obtained about the other.

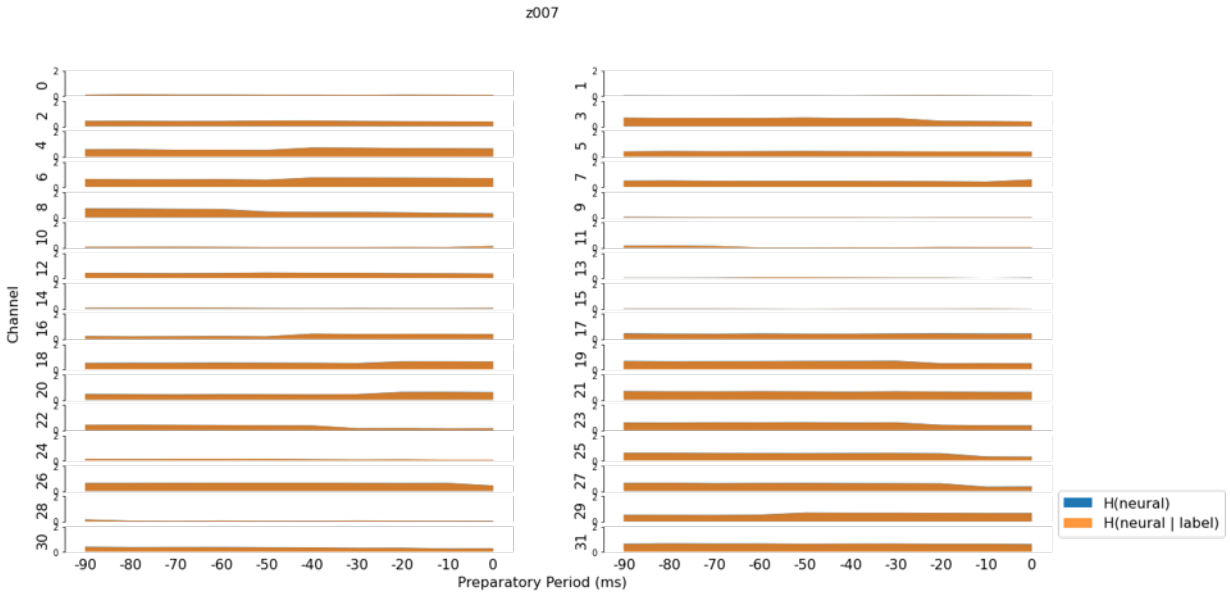

Figure 4.4: Entropy of spike counts $H(neural)$ and the conditional entropy of spike counts given call labels $H(neural|label)$ across channels computed every 10ms, up to 100ms before the onset of calls in z007. The difference, $H(neural) - H(neural|label)$ is the mutual information between spike counts and the type of call.

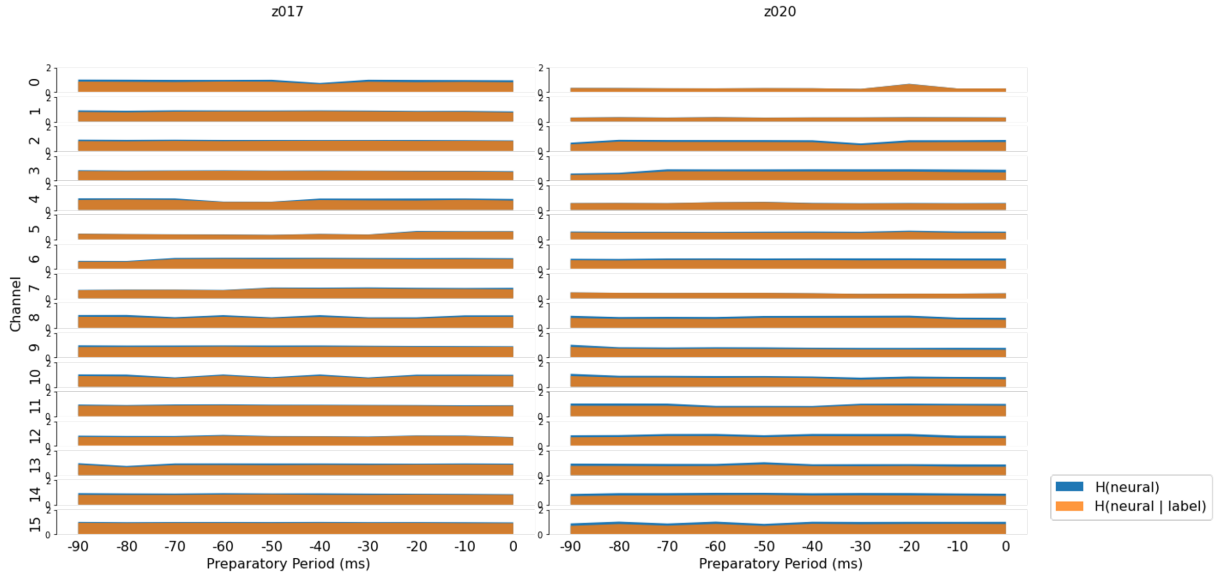

Figure 4.5: Entropy of spike counts $H(neural)$ and the conditional entropy of spike counts given call labels $H(neural|label)$ across channels computed every 10ms, up to 100ms before the onset of calls in z017 and z020. The difference, $H(neural) - H(neural|label)$ is the mutual information between spike counts and the type of call.

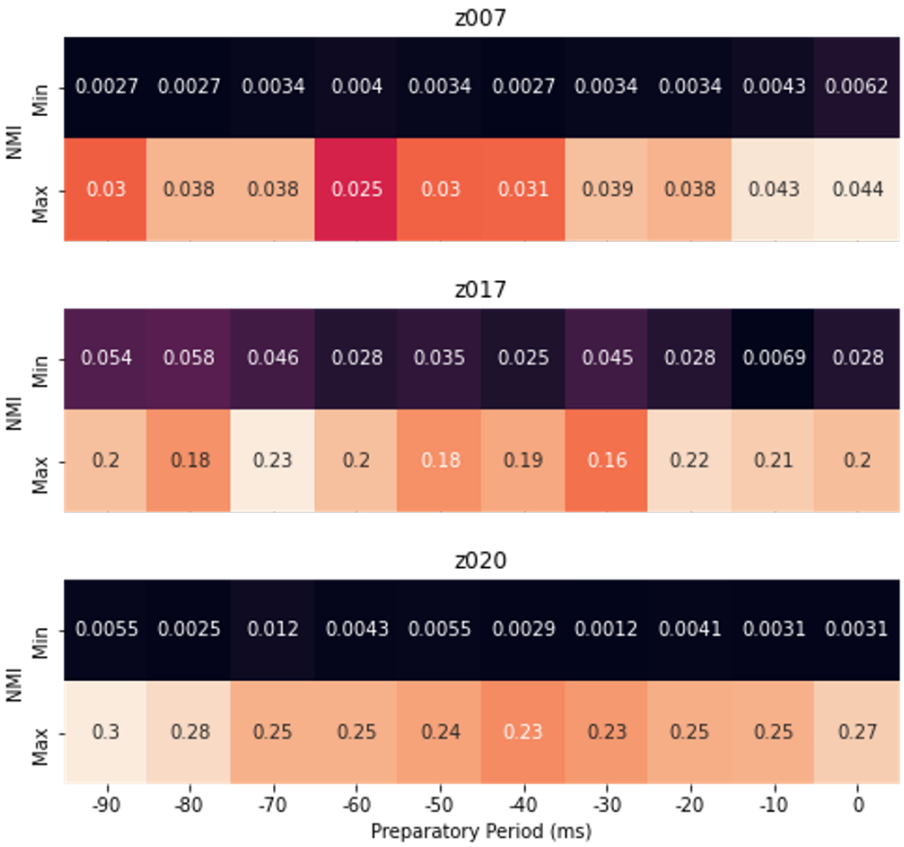

Figure 4.6: Minimum and maximum normalized mutual information (NMI) between the spike counts and the type of call across all channels computed every 10ms, up to 100ms before the onset of calls in z007, z017 and z020.

4.3.3 Gaussian Classification of Spike Counts

As the NMI analysis revealed that there exists some mutual information between the spike counts and the call labels, a Bayesian classifier was set up to determine the discriminability between the spike counts for different types of calls. For this, the spike counts across all channels at a particular preparatory time bin were modeled as a multivariate Gaussian. The call types were assumed to have uniform prior and the posterior probabilities were obtained as follows,

$$P(label|neural) = \frac{P(neural|label)P(label)}{P(neural)}$$

where $P(neural|label) \sim N(\mu_{C_i}, \Sigma_{C_i})$ for each label i. $P(label) = 0.5$ for all the zebra finches including z007, for whom the covariance for C_0 and C_1 were found to be non-invertible. Consequently, the calls belonging to that type couldn't be considered for classification. $P(neural) = \sum_{i \in labels} P(neural|i)$.

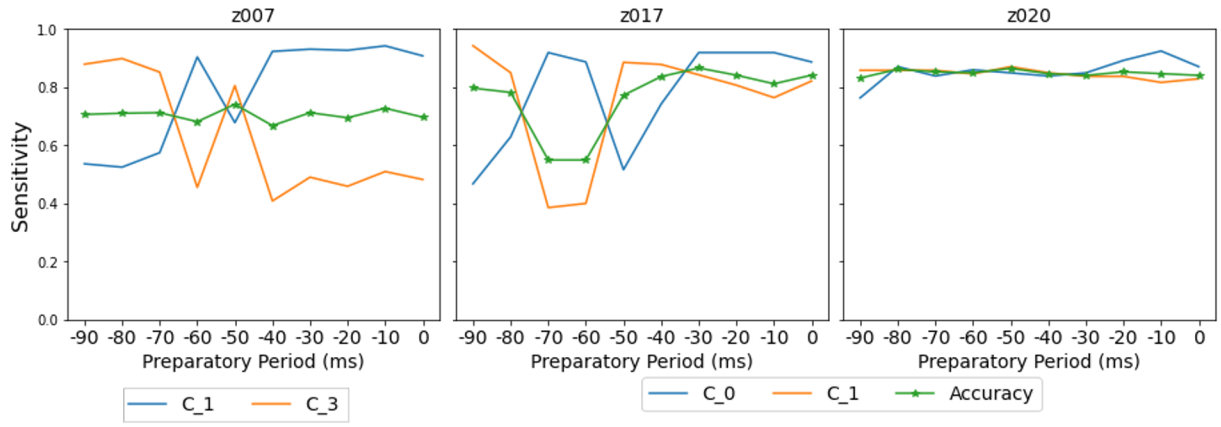

Figure 4.7: Sensitivity to the types of call and the accuracy of a multivariate Gaussian classifier for z007, z017 and z020, modelled using the spike counts computed every 10ms, upto 100ms before the onset of calls.

Figure 4.7 shows the sensitivity of the Bayesian classifier to call labels and its accuracy at different bins of time in the preparatory period. For z007, it is observed that from -30ms to onset of vocalization, the classifier is more sensitive to C_1 than C_0 whereas the sensitivity to both the call types is above 0.8 for z017 and z020. The overall classification accuracy is found to be

around 0.7, 0.8 and 0.82 for z007, z017 and z020 respectively, implying that the spike counts during the preparatory period [-20,0]ms are indicative of the call type that will be vocalized.

In conclusion, all the above analyses have shown that UMAP can primarily categorize calls into long and short calls. The spike counts for long calls are higher than those for short calls during the preparatory period that extends up to 20ms before the vocal behavior. NMI analysis revealed that there exists some mutual information between the spike counts and the call labels. Finally, a Gaussian classifier with a uniform prior can help determine the label of the spike counts with more than 70% accuracy.

Chapter 5

Conclusion

This thesis has addressed three major components that are crucial in developing neurally driven vocal prostheses for songbirds. These components are the automatic labeling of vocalizations in free behavior, the study of their latent representations and the information encoded in the neural activity of non-stereotypical behavior like calls. For each component, experiments were performed on an in-house dataset collected from adult male zebra finches.

Reviewing the deep learning approaches that have been leveraged for automatic labeling, the TweetyNet model, consisting of convolutional layers coupled to a recurrent layer with bidirectional LSTM units, was found to be highly sensitive to vocalizations. It showed high temporal precision in the detection of onset and offset of vocalizations. In fact, transient misclassifications within vocal events were close to 0ms. The model achieved maximum performance with less than 2 minutes of hand-labeled training data thereby significantly reducing the human effort involved in manual labeling. Transfer learning experiments revealed that the similarities within intra-species vocalizations can be leveraged to reduce training costs. A model based on the WaveNet architecture was also proposed, which was purely convolutional and operated directly on audio recordings thereby eliminating the need for feature engineering. Its sensitivity to vocalizations was neither too high nor too low, but it was highly precise in detecting the onset and offset of

vocalizations. Being a larger model with input that is typically recorded at high sampling rates, it is more data-intensive but faster than an equivalent model with recurrent layers. The capabilities of the WaveNet model need to be explored with augmented data, and the overall performance of the labeling system can be improved with a post-processing stage for removing false positives.

For the latent representation of vocalizations, Vector Quantized Variational Autoencoder (VQVAE) was explored as a viable technique. The quality of learnt latent representations was tested with a two-stage approach which involved coupling a pretrained VQVAE with the LSTM and the fully connected layers of the TweetyNet model. It was found that the resultant labeler performed equivalently to the original TweetyNet, indicating that the latent vectors of VQVAE encoded the relevant information from vocalizations. An exemplar novel vocalization was generated from the weighted latent representations of two existing vocal events, which could then be used in psychophysical experiments on songbirds. However, for such applications, models that operate on raw audio without any feature engineering would eliminate any human bias in feature design. Further, spectrograms of audio cannot typically capture the rapid fluctuations in vocal behavior that are observed in songbirds. Therefore, future experiments need to be performed directly on raw audio and one such approach is to use a WaveNet-based variational autoencoder. Incorporating perceptual loss while training could further improve the quality of the reconstructed or generated audio.

Finally, the last component of the thesis delved into analyzing the neural activity corresponding to different types of calls. Using the Uniform Manifold Approximation and Projection algorithm, at least 2 types of calls were observed for each zebra finch in the 2D space. These types could be broadly referred to as long and short calls. The spike counts for long calls were found to be higher than those for short calls during the preparatory period that spanned up to 30ms before the onset of vocal behavior. The mutual information analysis revealed that there is a dependency between the spike counts and the call type. High classification accuracy was achieved by a multivariate Gaussian classifier that modeled the spike counts across channels during the

preparatory period. These experiments on calls have provided a starting point for further analysis of non-stereotypical behavior.

These techniques need to be further optimized for size and latency as one moves towards real-time neural decoding. The decoded sequence of vocalizations can be provided as auditory feedback for closed-loop experiments or train young zebra finches.

Bibliography

[1] Keith A Josephs, Joseph R Duffy, Edythe A Strand, Mary M Machulda, Matthew L Senjem, Ankit V Master, Val J Lowe, Clifford R Jack Jr, and Jennifer L Whitwell. Characterizing a neurodegenerative syndrome: primary progressive apraxia of speech. *Brain*, 135(5):1522–1536, 2012.

[2] E Gorobets, R Kulsharipova, and M Novak. Speech disorders in patients with cognitive impairment caused by neurogenerative diseases: an overview. *Journal of Language and Literature*, 7(2):177–180, 2016.

[3] Gerald J Canter. Speech characteristics of patients with Parkinson's disease: I. intensity, pitch, and duration. *Journal of speech and hearing disorders*, 28(3):221–229, 1963.

[4] Moira Mulligan, Joseph Carpenter, Joanne Riddel, Maureen Kenny Delaney, Gary Badger, Patricia Krusinski, and Rup Tandan. Intelligibility and the acoustic characteristics of speech in amyotrophic lateral sclerosis (ALS). *Journal of Speech, Language, and Hearing Research*, 37(3):496–503, 1994.

[5] Florent Bocquelet, Thomas Hueber, Laurent Girin, Christophe Savariaux, and Blaise Yvert. Real-time control of an articulatory-based speech synthesizer for brain computer interfaces. *PLoS computational biology*, 12(11):e1005119, 2016.

[6] Jonathan S Brumberg, Alfonso Nieto-Castanon, Philip R Kennedy, and Frank H Guenther. Brain–computer interfaces for speech communication. *Speech communication*, 52(4):367–379, 2010.

[7] Josh Chartier, Gopala K Anumanchipalli, Keith Johnson, and Edward F Chang. Encoding of articulatory kinematic trajectories in human speech sensorimotor cortex. *Neuron*, 98(5):1042–1054, 2018.

[8] Frank H Guenther, Jonathan S Brumberg, E Joseph Wright, Alfonso Nieto-Castanon, Jason A Tourville, Mikhail Panko, Robert Law, Steven A Siebert, Jess L Bartels, Dinal S Andreasen, Princewill Ehirim, Hui Mao, and Philip R Kennedy. A wireless brain-machine interface for real-time speech synthesis. *PloS one*, 4(12):e8218, 2009.

[9] Kristofer E Bouchard and Edward F Chang. Neural decoding of spoken vowels from human sensory-motor cortex with high-density electrocorticography. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6782–6785. IEEE, 2014.

[10] Miguel Angrick, Christian Herff, Emily Mugler, Matthew C Tate, Marc W Slutzky, Dean J Krusienski, and Tanja Schultz. Speech synthesis from ECoG using densely connected 3d convolutional neural networks. *Journal of neural engineering*, 16(3):036019, 2019.

[11] Gopala K Anumanchipalli, Josh Chartier, and Edward F Chang. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498, 2019.

[12] Janaki Sheth, Ariel Tankus, Michelle Tran, Nader Pouratian, Itzhak Fried, and William Speier. Generalizing neural signal-to-text brain-computer interfaces. *Biomedical Physics & Engineering Express*, 7(3):035023, 2021.

[13] Qinwan Rabbani, Griffin Milsap, and Nathan E Crone. The potential for a speech brain–computer interface using chronic electrocorticography. *Neurotherapeutics*, 16(1):144–165, 2019.

[14] Ned T Sahin, Steven Pinker, Sydney S Cash, Donald Schomer, and Eric Halgren. Sequential processing of lexical, grammatical, and phonological information within broca's area. *Science*, 326(5951):445–449, 2009.

[15] Stephanie Martin, Iñaki Iturrate, José del R Millán, Robert T Knight, and Brian N Pasley. Decoding inner speech using electrocorticography: Progress and challenges toward a speech prosthesis. *Frontiers in neuroscience*, 12:422, 2018.

[16] Erich D Jarvis. Evolution of vocal learning and spoken language. *Science*, 366(6461):50–54, 2019.

[17] Kristina Simonyan, Hermann Ackermann, Edward F Chang, and Jeremy D Greenlee. New developments in understanding the complexity of human speech production. *Journal of Neuroscience*, 36(45):11440–11448, 2016.

[18] Michael S Brainard and Allison J Doupe. What songbirds teach us about learning. *Nature*, 417(6886):351–358, 2002.

[19] Johan J Bolhuis, Kazuo Okanoya, and Constance Scharff. Twitter evolution: converging mechanisms in birdsong and human speech. *Nature Reviews Neuroscience*, 11(11):747–759, 2010.

[20] Allison J Doupe and Patricia K Kuhl. Birdsong and human speech: common themes and mechanisms. *Annual review of neuroscience*, 22(1):567–631, 1999.

[21] Takashi Morita, Hiroki Koda, Kazuo Okanoya, and Ryosuke O Tachibana. Birdsong sequence exhibits long context dependency comparable to human language syntax. *bioRxiv*, 2020.

[22] David S Vicario, Nasir H Naqvi, and Jonathan N Raksin. Sex differences in discrimination of vocal communication signals in a songbird. *Animal behaviour*, 61(4):805–817, 2001.

[23] Shikha Kalra, Vishruta Yawatkar, Logan S James, Jon T Sakata, and Raghav Rajan. Introductory gestures before songbird vocal displays are shaped by learning and biological predispositions. *Proceedings of the Royal Society B*, 288(1943):20202796, 2021.

[24] Robert C Berwick, Kazuo Okanoya, Gabriel JL Beckers, and Johan J Bolhuis. Songs to syntax: the linguistics of birdsong. *Trends in cognitive sciences*, 15(3):113–121, 2011.

[25] Ezequiel M Arneodo, Shukai Chen, Vikash Gilja, and Timothy Q Gentner. A neural decoder for learned vocal behavior. *bioRxiv*, page 193987, 2017.

[26] Daril EVan Brown, Jairo Ismahar Chavez, Derek Hung Nguyen, Adam Kadwory, Bradley Voytek, Ezequiel Arneodo, Timothy Quinlan Gentner, and Vikash Gilja. Local field potentials in a pre-motor region predict learned vocal sequences. *bioRxiv*, 2020.

[27] Tim Sainburg, Marvin Thielk, and Timothy Q Gentner. Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLoS computational biology*, 16(10):e1008228, 2020.

[28] Sven E Anderson, Amish S Dave, and Daniel Margoliash. Template-based automatic recognition of birdsong syllables from continuous recordings. *The Journal of the Acoustical Society of America*, 100(2):1209–1219, 1996.

[29] Yarden Cohen, David A Nicholson, and Timothy J Gardner. Tweetynet: A neural network that enables high-throughput, automated annotation of birdsong. *bioRxiv*, 2020.

[30] Ben Pearre, L Nathan Perkins, Jeffrey E Markowitz, and Timothy J Gardner. A fast and accurate zebra finch syllable detector. *PloS one*, 12(7):e0181992, 2017.

[31] Nathan Trouvain and Xavier Hinaut. Canary song decoder: Transduction and implicit segmentation with ESNs and LTSMs. 2021.

[32] M Yu Byron, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in neural information processing systems*, pages 1881–1888, 2009.

[33] John P Cunningham and M Yu Byron. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509, 2014.

[34] Silvia Pagliarini, Nathan Trouvain, Arthur Leblois, and Xavier Hinaut. What does the canary say? Low-dimensional GAN applied to birdsong. 2021.

[35] Tim Sainburg, Marvin Thielk, and Timothy Q Gentner. Latent space visualization, characterization, and generation of diverse vocal communication signals. *bioRxiv*, page 870311, 2019.

[36] Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[37] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[38] Jack Goffinet, Richard Mooney, and John Pearson. Inferring low-dimensional latent descriptions of animal vocalizations. *bioRxiv*, page 811661, 2019.

[39] Tim Sainburg, Marvin Thielk, and Timothy Gentner. Learned context dependent categorical perception in a songbird. In *Conference on Cognitive Computational Neuroscience*, pages 1–4, 2018.

[40] Marvin Thielk, Tim Sainburg, Tatyana Sharpee, and Timothy Gentner. Combining biological and artificial approaches to understand perceptual spaces for categorizing natural acoustic signals. In *Conference on Cognitive Computational Neuroscience*, pages 1–4, 2018.

[41] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[42] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017.

[43] Andries Ter Maat, Lisa Trost, Hannes Sagunsky, Susanne Seltmann, and Manfred Gahr. Zebra finch mates use their forebrain song system in unlearned call communication. *PloS one*, 9(10):e109334, 2014.

[44] H Blair Simpson and David S Vicario. Brain pathways for learned and unlearned vocalizations differ in zebra finches. *Journal of Neuroscience*, 10(5):1541–1556, 1990.

[45] D Vicario, J Raksin, N Naqvi, N Thande, and H Simpson. The relationship between perception and production in songbird vocal imitation: what learned calls can teach us. *Journal of Comparative Physiology A*, 188(11-12):897–908, 2002.

[46] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[47] Alexay A Kozhevnikov and Michale S Fee. Singing-related activity of identified HVC neurons in the zebra finch. *Journal of neurophysiology*, 97(6):4271–4283, 2007.

[48] Marc F Schmidt. Pattern of interhemispheric synchronization in HVc during singing correlates with key transitions in the song pattern. *Journal of neurophysiology*, 90(6):3931–3949, 2003.

[49] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti Hämäläinen. MEG and EEG data analysis with mne-python. *Frontiers in neuroscience*, 7:267, 2013.

[50] Sarah MN Woolley. Early experience shapes vocal neural coding and perception in songbirds. *Developmental psychobiology*, 54(6):612–631, 2012.

[51] Tim Sainburg. *timsainb/noisereduce: v1.0.1*. Zenodo, Jun 2019.

[52] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[53] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[55] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016.

[56] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[57] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[58] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[59] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[60] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[61] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243, 1984.

[62] Alec Wright and Vesa Välimäki. Perceptual loss function for neural modeling of audio systems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 251–255. IEEE, 2020.

[63] Ishwarya Ananthabhotla, Sebastian Ewert, and Joseph A Paradiso. Towards a perceptual loss: Using a neural network codec approximation as a loss for generative audio models. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1518–1525, 2019.

[64] JN Holmes. The JSRU channel vocoder. In *IEEE (Communications, Radar and Signal Processing)*, volume 127, pages 53–60. IET, 1980.

[65] Julie E Elie and Frederic E Theunissen. The vocal repertoire of the domesticated zebra finch: a data-driven approach to decipher the information-bearing acoustic features of communication signals. *Animal cognition*, 19(2):285–315, 2016.