AVN: A Deep Learning Approach for the Analysis of Birdsong

Therese M.I. Koch[1*], Ethan S. Marks[1], Todd F. Roberts[1*]

Department of Neuroscience, UT Southwestern Medical Center, Dallas TX, USA

Correspondence should be addressed to TFR and TMIK:

Todd.Roberts@utsouthwestern.edu, Therese.Koch@utsouthwestern.edu

Abstract

Deep learning tools for behavior analysis have enabled important new insights and discoveries in neuroscience. Yet, they often compromise interpretability and generalizability for performance, making it difficult to quantitively compare phenotypes across datasets and research groups. We developed a novel deep learning-based behavior analysis pipeline, *Avian Vocalization Network* (AVN), for the learned vocalizations of the most extensively studied vocal learning model species – the zebra finch. AVN annotates songs with high accuracy across multiple animal colonies without the need for any additional training data and generates a comprehensive set of interpretable features to describe the syntax, timing, and acoustic properties of song. We use this feature set to compare song phenotypes across multiple research groups and experiments, and to predict a bird's stage in song development. Additionally, we have developed a novel method to measure song imitation that requires no additional training data for new comparisons or recording environments, and outperforms existing similarity scoring methods in its sensitivity and agreement with expert human judgements of song similarity. These tools are available through the open-source AVN python package and graphical application, which makes them accessible to researchers without any prior coding experience. Altogether, this behavior analysis toolkit stands to facilitate and accelerate the study of vocal behavior by enabling a standardized mapping of phenotypes and learning outcomes, thus helping scientists better link behavior to the underlying neural processes.

Introduction

A deep understanding of animal behavior is fundamental to a deep understanding of the brain. However, accurate, quantitative description of animal behavior, particularly in ethologically relevant contexts, remains a substantial challenge in neuroscience research. In recent years, careful observation of motor and vocal behaviors is increasingly being replaced with machine learning and deep learning-based approaches. These tools allow researchers to consider much greater volumes of data than was previously possible, to uncover patterns in animal behavior that are undetectable to humans, and have led to important insights into ethologically relevant behaviors, and the effects of experimental interventions thereupon [1-3]. However, this increased power often comes at the expense of interpretability and generalizability.

An increasing number of supervised deep learning methods are being developed for the automated annotation of animal vocalization behavior [4-7], and unsupervised methods for dimensionality reduction and analysis [3, 8-11]. While unsupervised approaches are very powerful, and have been shown to explain more variance in vocalization repertoires than hand-selected acoustic features [8], the features that they

generate are notoriously difficult to interpret, and specific to the exact dataset from which they were derived [3, 8, 9]. As a result, unsupervised data-driven methods, while allowing detailed comparison of individuals within the same data set, make it more difficult to compare the nature and severity of vocal phenotypes across experiments and research groups.

To truly maximize the benefits of machine learning and deep learning methods for behavior analysis, their power must be balanced with interpretability and generalizability. This can be achieved by combining automated annotation with a carefully selected set of meaningful features, thereby creating a common feature space for the comparison of behavioral phenotypes across research groups, experimental conditions, and studies. For speed, ease of use, and standardization, the annotations should be generated without the need for any training data or hyperparameter setting for new individuals or recording conditions. The features should be consistent across recording conditions, allowing direct, meaningful comparisons between research groups. The feature set should be comprehensive, describing multiple aspects of the behavior. Finally, the features should be interpretable, allowing researchers to form concrete hypotheses about how different manipulations will affect specific features, and use observed feature values to guide future experimental design.

We have developed an analysis pipeline called *Avian Vocalization Network (AVN)* which satisfies these criteria for zebra finch song analysis. Zebra finches are the most popular animal model for the study of vocal learning. They learn to sing a single, highly stereotyped song by memorizing the song of an adult tutor, then refining their vocalizations to match this song template during a sensorimotor learning period early in development (Fig 1a); a process which bears many parallels to human speech learning [12]. Typical zebra finch songs consist of a variable number of introductory notes, followed by multiple repetitions of a motif, composed of 3 to 10 unique syllable types produced in a stereotyped sequence. Traditionally, zebra finch song has been analyzed by segmenting the song into syllables, then manually labeling syllables based on visual inspection of their spectrograms [13-15]. This process is very labor intensive, which limits the number of songs that can be considered at a time. Manual syllable labeling and motif identification can also be subjective and therefore susceptible to experimenter bias, as motif composition and syllable types can be somewhat ambiguous, particularly in young birds with immature song and in birds with experimentally disrupted songs.

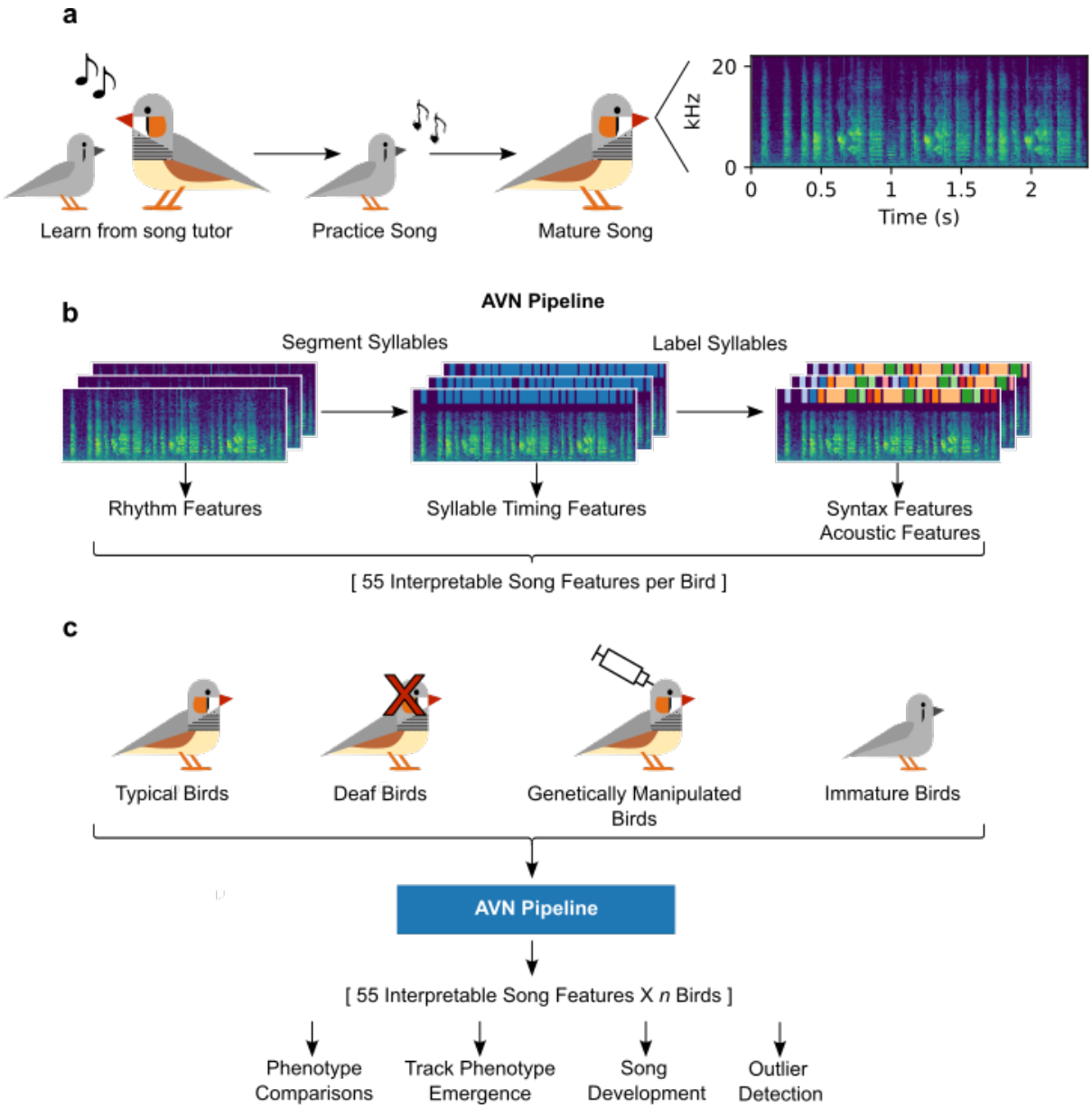

Figure 1 – Overview of AVN song analysis pipeline. a. Schematic timeline of zebra finch song learning. **b.** Overview of AVN song analysis pipeline. Spectrograms of songs are automatically segmented into syllables then syllables are labeled. The raw spectrograms are used to calculate features describing the rhythm of a bird's song, the segmentations are used to calculate syllable-level timing features, and the labeled syllables are used to calculate syntax-related features and acoustic features of a bird's song. **c.** Birds from different research groups, with multiple different song phenotypes can all be processed by the AVN pipeline, generating a matrix of directly comparable, interpretable features, which can be used for downstream analyses including phenotype comparisons, tracking the emergence of a phenotype over time, investigating song development, and detecting individual outlier birds with atypical song phenotypes.

We tested two deep-learning approaches for syllable segmentation, which don't require any additional training data or hyperparameter setting for new birds. We then applied unsupervised dimensionality reduction and clustering methods to assign labels to these automatically segmented syllables. Finally, we use the resulting annotated songs to calculate a set of 55 interpretable features which

describe the syntax, timing, and acoustic properties of a set of songs (Fig 1b). We show that the automated annotation performs consistently well across multiple zebra finch colonies, and that the feature set can be used to glean mechanistic insights from the comparison of vocal phenotypes, and to predict a bird's stage in song development (Fig 1c). We also developed a new method to compare two birds' syllable repertoires in order to measure song learning, which outperforms existing song similarity scoring methods on multiple key metrics. The complete pipeline is available as an open-source python package and as an application with a graphical user interface, allowing researchers with no prior coding experience to easily annotate their songs, calculate the feature set, and calculate song similarity scores (supplemental Fig 1).

Results

Comparing deep learning methods for fully automated syllable segmentation

To accurately segment and label zebra finch songs without the need for any individual-specific training data or hyperparameter tuning, we tested and compared two different deep-learning based approaches for syllable segmentation. Traditionally, zebra finch song is segmented based on an amplitude threshold [10, 16]. The best value for this amplitude threshold depends heavily on recording conditions and background noise levels, and setting this threshold often requires careful trial and error by a human annotator. Amplitude-based segmentation methods also cannot distinguish between song syllables and noises, like wing flaps or other non-vocal artifacts, which can contaminate downstream analyses. Instead of relying on amplitude alone, we compared two deep learning models, TweetyNet [5] and WhisperSeg [6], which take the full spectral content of the audio into account when performing segmentation.

We tested these two segmentation methods with a dataset of over 1000 manually annotated songs from 35 adult zebra finches, including birds with typical song production, isolate birds raised without a song tutor, and birds with disrupted song production due to knockdown of the transcription factor FoxP1. TweetyNet was designed to simultaneously segment and label syllable types, by assigning syllable labels to short spectrogram frames. This application requires re-training for each individual bird, so we instead trained TweetyNet to label spectrogram frames as simply containing vocalizations, silence, or noise. We trained it with 34 of the 35 birds in the dataset and evaluated segmentation accuracy with the remaining bird, repeating this once for each bird in the dataset. This allows the model to learn an abstract notion of vocalization vs. non-vocalization which generalizes well to new individuals not included in training. The WhisperSeg model is already trained for segmentation of new individuals, so we used the existing standard model to segment each of our birds. Segmentation accuracy was evaluated against expert human annotations by calculating the precision, recall and F1 scores of syllable onset detections within 10ms of a syllable onset in the manual annotations.

WhisperSeg shows the best performance, with a mean F1 score of 0.882(+- SEM 0.02), compared to TweetyNet's score of 0.824 (+- SEM 0.03) and a simple amplitude segmentation algorithm (RMSE) with

a mean score of 0.593 (+-SEM 0.02) (Fig 2a). WhisperSeg's precise onset times were also more consistent with expert human annotations than both other methods (median absolute time difference of 1.75ms for WhisperSeg, 2.22ms for TweetyNet, and 3.81ms for RMSE) (Fig 2b). All 3 methods performed similarly for the typical, isolate and FP1 KD birds (supplemental 2 a-f). As a further test of the generalization of these methods, we applied them to a dataset of manually annotated songs from 25 birds from the Rockefeller University Field Research Center Song Library [17]. Using the pre-trained WhisperSeg model, and a TweetyNet model trained on all 35 birds from the UTSW colony and none from the Rockefeller Song Library, each of these models yielded very similar segmentation accuracy scores to those obtained with the UTSW colony (Fig 2a,c, supplemental 3).

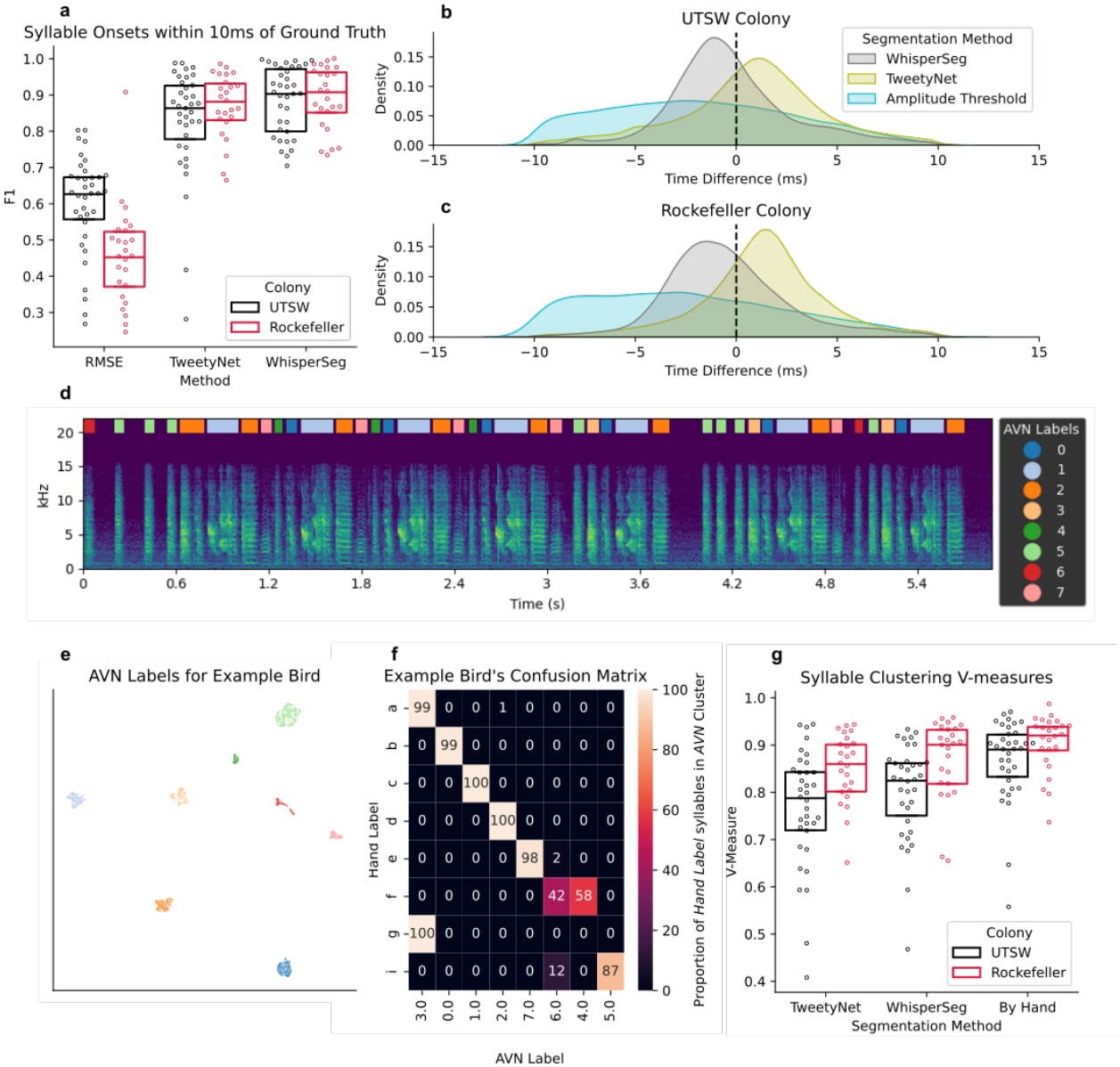

Figure 2 – Automated syllable annotation metrics. a. F1 scores for syllable onset detections within 10ms of a syllable onset in the manual annotations of each bird (n=35 from UTSW and n=25 from Rockefeller) across segmentation methods. **b.** Distribution of time-differences between predicted syllable onsets and their best matches in the manual annotation, across segmentation methods. Distributions include all matched syllables across all 35 birds from the UT Southwestern colony (UTSW) and (**c.**) 25 from Rockefeller. **d.**

Example spectrogram of a typical adult zebra finch. The song was segmented with WhisperSeg and labeled using UMAP & HDBSCAN clustering. Colored rectangles reflect the labels of each syllable. **e.** Example UMAP plot of 3131 syllables from the same bird as in *d* and *f*. Each point represents one syllable segmented with WhisperSeg, and colors reflect the AVN label of each syllable. **f.** Example confusion matrix for the bird depicted in *d* and *e*. The matrix shows the percentage of syllables bearing each manual annotation label which fall into each of the possible AVN labels. **g.** V-measure scores for AVN syllable labels compared to manual annotations for each bird (n=35 from UTSW and n=25 from Rockefeller), across segmentation methods.

Accurate, fully unsupervised syllable labeling

Next, we assign syllable labels to these segmented units. To achieve this, we first performed UMAP dimensionality reduction [18] on spectrograms of the segmented syllables, then performed hierarchical density based clustering (HDBSCAN) [19] on the syllables' UMAP embeddings, as in [10]. We calculated the UMAP embeddings of all segmented syllables for each of the 35 birds in our dataset using manual segmentations, WhisperSeg, or TweetyNet segmentations. In all cases, we found that the syllables formed multiple dense clusters, which corresponded well to manually annotated syllable labels (Fig 2 d-g, supplemental Fig 4).

Using manual segmentation yielded the best agreement with manual labels (mean v-measure score = 0.87 +- 0.01) which is to be expected, as no discrepancies are introduced during segmentation (Fig 2g, supplemental 5). When clustered, WhisperSeg's segments yielded better agreement with manual labels than TweetyNet's (WhisperSeg mean v-measure = 0.80 +- 0.02, TweetyNet's mean v-measure - 0.77 +- 0.02). We observe similar performance on our second dataset of 25 typical adult zebra finches from the Rockefeller Song Library, suggesting that these methods generalize well across colonies and recording environments (Fig 2g).

Altogether, we conclude that WhisperSeg followed by UMAP-HDBSCAN clustering produces the most accurate syllable labels. These will be referred to as AVN labels henceforth in this manuscript. AVN labels are produced without the need for any per-bird parameter tuning or model training. This approach not only saves experimenters time when analyzing many birds, but also reduces the potential for experimenter bias during song annotation. AVN labeling generalizes well across multiple zebra finch colonies, suggesting that it can be easily adopted by new research groups without the need for extensive additional validation. Thus, we hope it can serve as a new standard for song annotations when manual annotation is not required.

Analyzing Song Syntax

The automatically generated AVN labels can be used to visualize and quantify a bird's song syntax. Typical zebra finches produce syllables in a very predictable order, where the syllable type that a bird will sing can be reliably predicted based on the immediately preceding syllable type [15, 20]. Many studies have found that manipulations of the neural circuitry underlying song learning and production can disrupt a bird's

syntax, leading to more variable sequences [15, 21-23]. Methods used to quantify these syntax disruptions vary across papers and research groups, making it impossible to directly compare the severity of disruptions. We propose a comprehensive suite of features to describe a bird's song syntax, which can all be calculated using the AVN python package and AVN graphical application.

First, we developed a new song syntax visualization, called a *syntax raster plot* which lets researchers view a large number of song bouts' syllable sequences simultaneously (Fig 3a). We can also visualize syntax using a transition matrix, which gives the probability of a syllable type being produced, given the preceding syllable type (Fig 3b). We quantify the stereotypy of a bird's syntax by calculating the entropy rate of the transition matrix, and find a strong correlation ($r = 0.89$, $p < 0.05$) between entropy rates calculated using AVN labels and manual annotations, showing that our AVN labels are sufficiently reliable to describe a bird's syntax stereotypy (Fig 3c). We also find the same statistical relationship between groups as with manual annotations, namely that birds with FP1 KD and isolate birds have significantly higher entropy rates than typical birds (One way ANOVA $F(2, 32) = 15.05$, Tukey HSD p-adj FP1 vs. typical < 0.005, p-adj isolate vs. typical <0.005) (Fig 3d). Multiple studies have also found that neural song-circuit manipulations can induce a 'stutter' in birds, i.e. increase the rate of syllable repetitions in their songs [24-26], so we've introduced two additional metrics to specifically look at the rate of syllable repetitions in a bird's song; the mean number of times a syllable is produced in a row each time it is sung (repetition bout length), and the CV of the number of syllable repetitions (CV repetition bout length) (supplemental 6e-f).

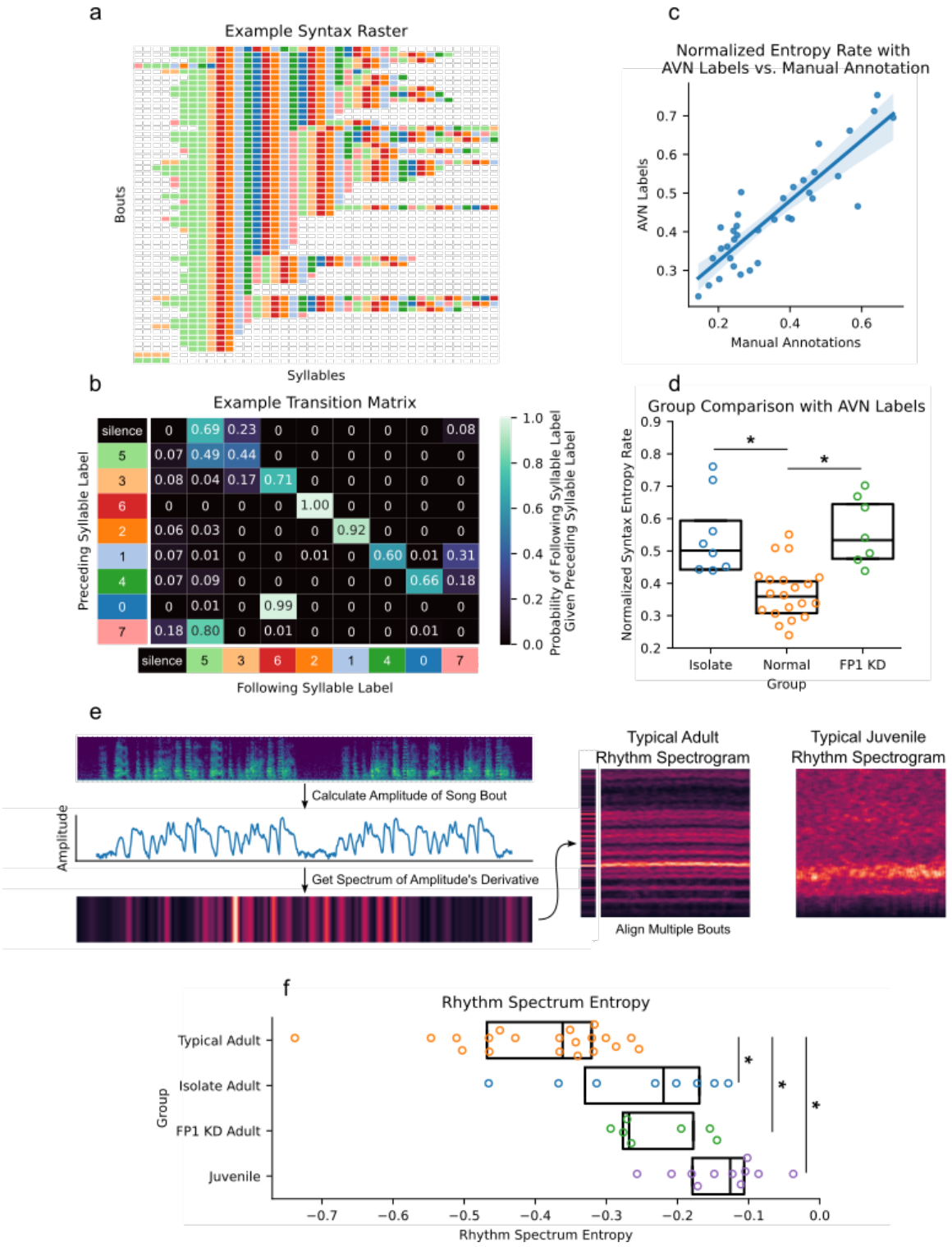

Figure 3 – Song syntax and timing analysis with AVN. a. Example syntax raster plot for a typical adult zebra finch made with AVN labels. Each row represents a song bout, and each colored block represents a syllable, colored according to its AVN label. **b.** Example transition matrix from the bird featured in *a*. Each cell gives the probability of the bird producing the 'following syllable', given that they just produced a syllable with the 'preceding syllable' label. **c.** Correlation between normalized entropy rate scores calculated for each bird using manual annotations or AVN labels (n=35 birds from UTSW, r = 0.89, p<0.005). **d.** Comparison of normalized entropy rates calculated with AVN labels across typical (n=20), isolate(n=8), and FP1 KD (n=7) adult zebra finches (One Way ANOVA F(2, 32) = 15.05, p <0.005, Tukey HSD * indicates p-adj < 0.005). **e.** Schematic representing the generation of rhythm spectrograms. The amplitude trace of each song file is calculated, then the spectrum of the first derivative of the amplitude trace is computer. The spectra of multiple song

files are concatenated to form a rhythm spectrogram, with bout index on the x-axis and frequency along the y axis. The example rhythm spectrograms show the expected banding structure of a typical adult zebra finch, and the less structured rhythm of a typical juvenile zebra finch (50dph). **f.** Comparison of rhythm spectrum entropies cross typical (n=20), isolate (n=8), FP1 KD (n=7) adult zebra finches (>90dph), and juvenile zebra finches (n = 11, 50-51dph) (One Way ANOVA $F(3, 43) = 17.0$, $p < 0.05$, Tukey HSD * indicates p-adj < 0.05).

Analyzing Song Timing

Song timing can refer to the durations of individual syllables and gaps, or to the rhythmic patterns of a song bout. We have developed and validated multiple metrics to describe song timing at each of those scales, which can easily be calculated using the AVN python package or graphical application. First, we look at the timing of individual syllables and gaps by plotting the distribution of their durations based on our WhisperSeg segmentations. Typical mature zebra finches have very stereotyped syllable durations across renditions of the same syllable type, which result in a distribution of syllable durations consisting of multiple narrow peaks, each corresponding to a different syllable type (supplemental Fig 7b). Immature birds, on the other hand, have very variable syllable durations, and tend to have a single broad peak and long positive tail in their syllable duration distributions [27, 28]. We observe these same patterns using our WhisperSeg segmentations when we apply them to our dataset of 35 mature birds, and an additional 11 juvenile birds aged 50-51 days post hatch (dph). We quantify the maturity of a bird's syllable timing by calculating the entropy of their syllable duration distribution, which will approach 1 when density is evenly spread across syllable durations (as in juvenile birds), and approach 0 when density is concentrated in a narrow range of syllable durations, as was done in [28]. Indeed, using our WhisperSeg segmentation, we find that juvenile birds have significantly higher syllable duration entropies than adult birds ($F(3, 43) = 17.43$, $p<0.005$, Tukey HSD p-adj juvenile vs. typical adult <0.05) (supplemental Fig 7e). The syllable duration entropy values that we obtain with WhisperSeg segmentation are also highly correlated with those scores that we obtain from manual segmentation ($r = 0.85$, $p<0.05$) (supplemental Fig 7a, c), further indicating that our automated segmentation is sufficiently accurate for downstream analyses.

In addition to syllable level timing, we have implemented multiple metrics describing a song's rhythm at the bout level based on the 'rhythm spectrogram' first proposed in [29]. A rhythm spectrum is constructed by taking the Fourier transform of the derivative of the amplitude trace of a song. If the song consists of multiple motifs with a consistent rhythm, the song's amplitude will have a repeating fluctuation pattern, which will be reflected in its spectrum. By calculating the 'rhythm spectrum' of multiple bouts, then concatenating them into a rhythm spectrogram, we can get a clear impression of a bird's overall rhythmicity, and the consistency of that rhythm across song bouts (Fig 3e). We quantify the strength of this rhythm by computing the Wiener Entropy of the mean rhythm spectrum, and we quantify the consistency of the rhythm across bouts by calculating the coefficient of variation of the peak frequency across each bout in the rhythm spectrogram (supplemental Fig 8). We find that juvenile birds have significantly higher rhythm spectrum entropies (Figure 3f, One Way ANOVA $F(3, 43) = 17.0$, Tukey HSD juvenile vs. typical adult p-adj < 0.005)

and higher peak frequency CVs than adult birds (supplemental Fig 8, One Way ANOVA $F(3, 43) = 8.23$, Tukey HSD juvenile vs. typical adult p-adj < 0.05). Whereas the FP1 KD birds' syllable duration entropies are squarely in line with typical adults (supplemental Fig 7e, Tukey HSD FP1 KD vs typical adult p-adj = 0.53), their rhythm spectrum entropies (Fig 3f) and gap duration entropies (supplemental Fig 7f) are significantly higher (Tukey HSD FP1 KD vs. typical adult p-adj <0.05). This is consistent with our earlier finding that the FP1 KD birds have more variable syllable sequencing, and also highlights the complementary nature of these metrics; when considered together, they provide a comprehensive overall description of a bird's song production.

Comparing Song Disruptions with AVN Features

In addition to syntax and timing features, AVN can also calculate a suite of acoustic features, including goodness of pitch, mean frequency, frequency modulation, amplitude modulation, entropy, amplitude, and pitch. This feature set is well established for describing zebra finch song, thanks to the Sound Analysis Pro application [30]. These features are calculated for each frame of a spectrogram, but to facilitate comparisons between birds, we take the mean value of each feature for every syllable rendition, then compute the overall mean value and the coefficient of variation across renditions of the same syllable type. We then select the syllable types with the minimum, median and maximum values with respect to each feature to represent the overall acoustic properties of a bird's song. This results in a total of 48 acoustic features for each bird. When combined with our 3 syntax related features and 4 timing related features, we are left with a complete set of 55 features to describe all major aspects of a bird's song production. This feature set represents an extremely valuable resource for comparing experimental groups, for tracking song phenotypes over time, or for detecting birds with atypical song production.

To showcase the AVN feature set's potential for comparing birds across experiments and research groups, we calculated this feature set for 53 typical adult zebra finches, 16 isolate-reared zebra finches, and 7 FP1 KD zebra finches from the UTSW colony, as well as 25 typical adult zebra finches from the Rockefeller Song Library [17], and 4-sham deafened birds and 5 early-deafened birds from Hokkaido University, originally recorded for [31]. We fit a Linear Discriminant Analysis (LDA) model to a dataset containing only typical and isolate zebra finches and achieved a 95% classification accuracy between these two groups (Figure 4a), with the most important features being higher syntax entropy rates, higher syllable duration variance, and higher rhythm entropies for isolates compared to typical birds (supplemental Fig 9a). We repeated this process for typical hearing and deaf birds and achieved a 99% classification accuracy (Fig 4b), with the most important features being higher mean frequency variance, lower absolute mean frequency, and higher syllable duration variance for deaf birds compared to hearing birds (supplemental Fig 9b).

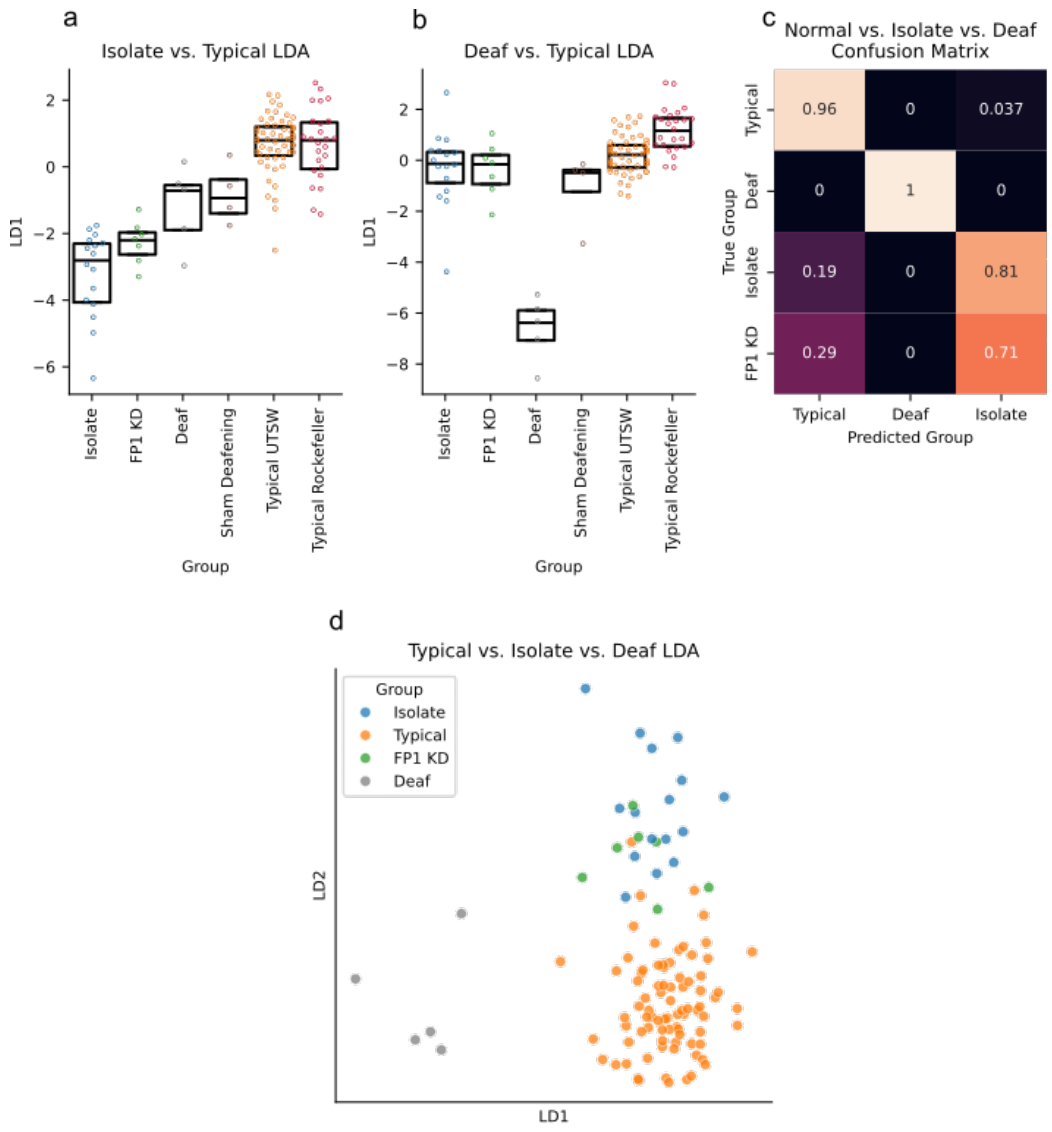

Figure 4 – Song phenotypes classification with AVN features. a. Linear discriminant values for multiple groups of birds generated from a model trained to discriminate between typical and isolate zebra finches (n=16 isolate birds, 7 FP1 KD birds, 5 deaf birds, 4 sham deafening birds, 53 typical zebra finches from the UTSW colony and 25 typical zebra finches from Rockefeller). **b.** Linear discriminant values for multiple groups of birds generated from a model trained to discriminate between typical and deaf zebra finches. Same birds as in *a*. **c.** Confusion matrix indicating the LDA model's classification of typical, deaf, isolate and FP1 KD birds from aa model trained to discriminate between typical, deaf, and isolate birds. Scores for typical, deaf, and isolate birds were obtained using leave-one-out cross validation, and FP1 KD scores were obtained using a model fit to all typical, deaf and isolate birds. **d.** Plot of the linear discriminant coordinates of isolate (n=16), typical (n=78), and FP1 KD birds (n=7) for a model trained to discriminate between typical, deaf, and isolate birds. FP1 KD birds overlap most with isolate birds in this LDA space, indicating that their song production most closely resembles that of isolates.

Finally, we fit an LDA model to a dataset containing typical, isolate, and deaf zebra finches, and used this model to assess which of these groups the FP1 KD birds most closely resemble. Previous work suggests that FP1 KD in the targeted brain region impairs tutor song memory formation while leaving song motor learning intact [21]. Thus, we would expect the FP1 KD birds' songs to most closely resemble isolates', who

have no tutor song memory but do have access to auditory feedback of their own vocalizations, and to not resemble deaf birds who have neither tutor song memories, nor access to auditory feedback. Indeed, we find that 5/7 FP1 KD birds are classified as isolates by the LDA classifier, with the other 2/7 being classified as typical birds (Fig 4c,d). This supports our hypothesis about the nature of the song disruption in FP1 KD birds and highlights the utility of a common feature set for comparing song phenotypes. We hope that the ease of calculation and completeness of this feature set will facilitate phenotypic comparisons across the field of songbird neuroscience, particularly as the field grows in its ability to conduct genetic manipulations of the neural circuits involved in different aspects of song learning and production.

Tracking Song Development with AVN Features

To further showcase the potential of these AVN features for zebra finch song analysis, we also used them to track song development. We calculated the AVN feature sets for 14 birds from UTSW and 5 birds from Duke University [9] at multiple timepoints during song development, ranging from 46 dph to 102 dph. We fit a Generalized Additive Model (GAM) to predict a bird's age based on its AVN features and find that we achieve the most accurate age predictions with a model that considers a bird's syllable duration entropy, their syntax entropy, absolute syllable durations, and the variability of goodness of pitch, syllable duration and Weiner entropy across renditions (Fig 5b). When trained with data from all but one bird and tested on the remaining bird, this model can predict a bird's age within 7 dph for 50% of age points, and within 11 dph for 75% of age points (Fig 5a). Its performance is best for younger birds, with prediction accuracy dropping considerably for birds over 80 dph, which is expected as song changes slow with age and eventually stabilize when birds reach around 90 dph.

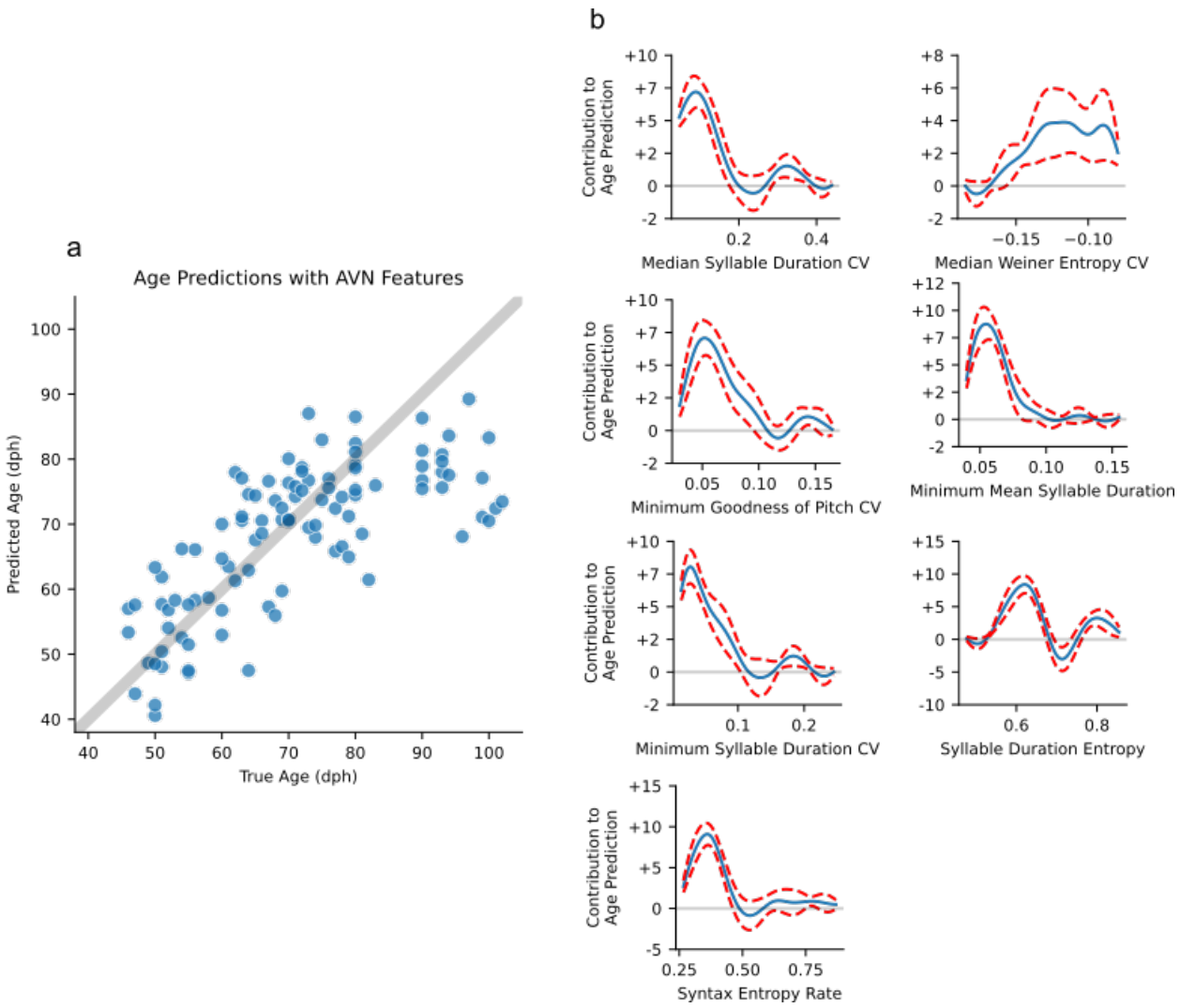

Figure 5 – Age prediction with AVN features. a. Generalized additive model's age predictions vs. true ages for 103 days of song recordings across 19 individual birds. Model predictions were generated using leave-one-bird-out cross validation. The grey line indicates where points would lie if the model were perfectly accurate. **b.** Partial dependence functions for each feature in the GAM model. The values of each feature along the x-axis map onto learned contributions to the age prediction along the y-axis. The GAM model's prediction is the sum of these age contributions based on each day of song's feature values, plus an intercept term.

Measuring Song Imitation

So far, we've demonstrated how AVN's features can be used to describe and compare adult and juvenile song production across experiments and research groups. While these features are sufficient to predict a bird's song learning stage and to detect abnormalities in experimental groups, they don't directly reflect song learning success. Zebra finches learn song by imitating an adult tutor, and this song learning is typically assessed by comparing a pupil bird's song to its tutor's, with higher similarity reflecting more successful learning ([30], but see [17]). Many methods for zebra finch song similarity scoring currently exist, however they all require either the manual identification of pupil and/or tutor motifs [30, 32, 33], which limits the number of renditions that can be considered and has the potential to introduce experimenter bias, or

require retraining or re-calibration when applied to new tutor-pupil pairs [8, 34], which makes it impossible to directly compare learning outcomes across experiments.

To overcome these limitations, we have developed a novel similarity scoring system which doesn't require any manual motif identification, or any retraining or re-calibration for new tutor-pupil comparisons. Our approach involves a deep convolutional neural network which is trained with a dataset of over 16,000 manually annotated syllables from 21 adult zebra finches from the UTSW colony. These syllables are presented to the model in triplets, consisting of a randomly selected 'anchor' syllable, a 'positive' syllable which belongs to the same type as the anchor, and a 'negative' syllable, which belongs to a different syllable type. The model learns to map spectrograms of syllables to an 8-dimensional embedding space, such that the anchor syllable's embedding is closer to the positive's embedding than to the negative's. We use the trained network to compute the syllable embeddings for hundreds of syllables produced by a pupil bird and by its tutor and measure the similarity between their songs by calculating the Earth Mover's Distance (EMD) between their syllable distributions (Fig 6a).

We first validated this approach with a dataset of 30 typical tutor-pupil pairs from the UTSW colony segmented using WhisperSeg, none of whom share a song tutor with any of the birds used to train the model. The model consistently yields higher EMD dissimilarity scores between a pupil and unrelated bird, compared to a pupil vs. another bird with the same tutor (a 'sibling') and a pupil vs. its tutor, as expected (Fig 6b). Across each of these comparisons our method produces EMD scores following the same pattern and in the same absolute range for the UTSW dataset and for a dataset of 25 tutor-pupil pairs from the Rockefeller Song Library, despite these birds being recorded under different conditions from any of the birds used in model training (Fig 6b). This shows that the trained model generalizes well to birds from other research groups without the need for any additional fine tuning, and thus can serve as a standard approach for the entire field. Our approach outperforms Sound Analysis Pro [30] (contrast index = 0.156) [32] and Mandelblat-Cerf & Fee 2014 (contrast index = 0.41) [32], based on its 'contrast index', and yields similarly high contrast indices for both the UTSW and Rockefeller datasets (Supplemental 10c, UTSW mean contrast index = 0.521 +- 0.019, Rockefeller mean contrast index = 0.548 +- 0.017, t-test p = 0.30). EMD scores produced by this model also agree better with expert human judgements of song similarity than do %similarity scores calculated with Sound Analysis Pro (Figure 6c, EMD vs. human expert absolute r = 0.87, Supplemental 10b, SAP %similarity vs. human expert absolute r = 0.33). In fact, the correlation between EMD scores and our human expert panel's scores is within the range of correlations of individual experts with the mean of the remaining evaluators, indicating that our method is as reliable as an individual human expert (supplemental Fig 10d).

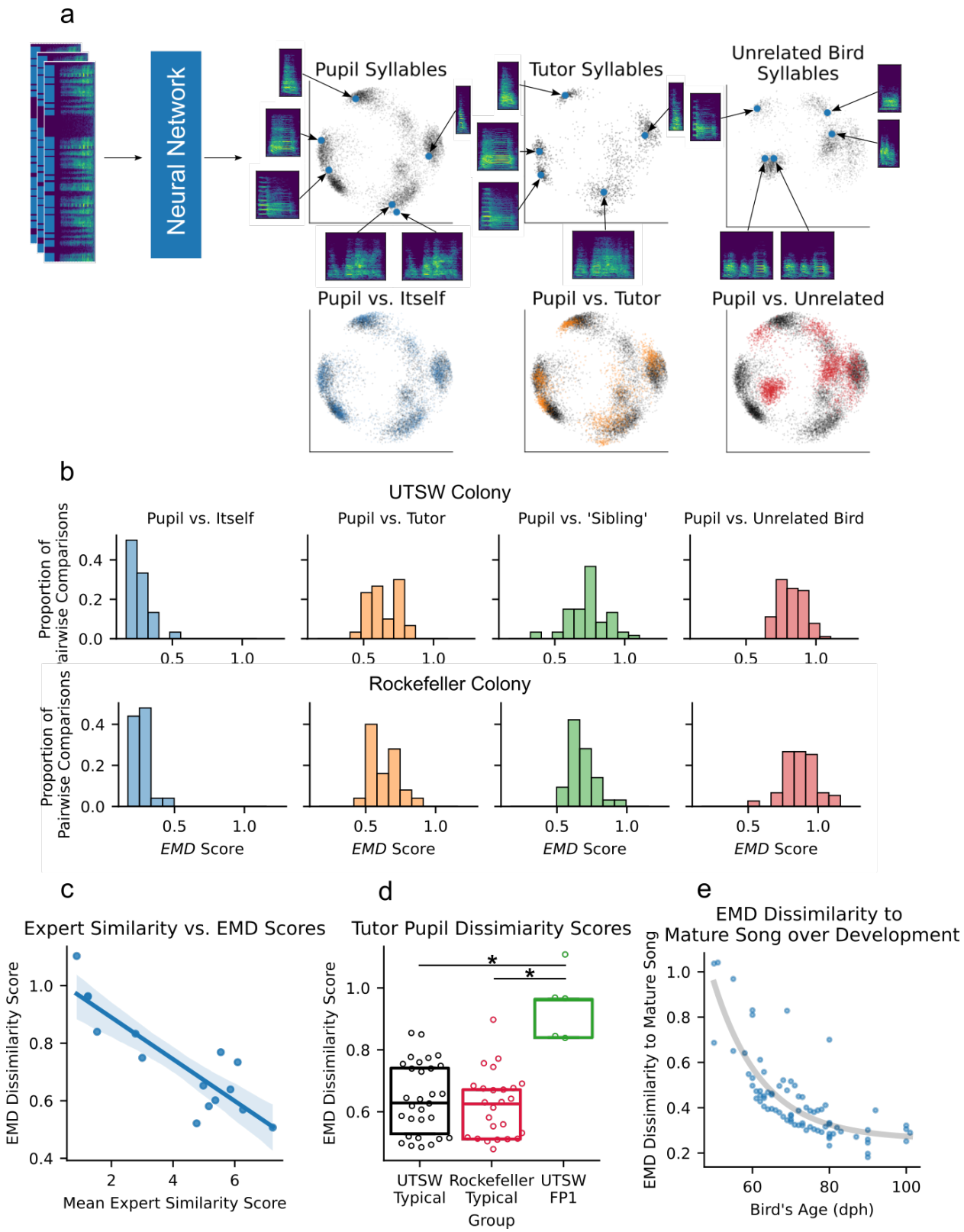

Figure 6 – Illustration and validation of AVN's song similarity scoring method. a. Schematic of the similarity scoring method. A deep convolutional neural network is used to embed syllables in an 8-dimensional space, where each syllable is a single point, and similar syllables are embedded close together. The first 2 principal components of the 8-dimensional space are used for visualization purposes only here. The syllable embedding distributions for two random subsets of syllables produced by the same pupil on the same day have a high degree of overlap. The distributions of all syllables from a pupil and his song tutor are less similar than a pupil compared to himself, but still much more similar than a pupil and a random unrelated bird. **b.** Earth Mover's Distance (EMD) dissimilarity score distribution for

comparisons between a pupil and itself (n=30 comparisons for UTSW, n = 25 for Rockefeller), a pupil and its tutor (n=30 comparisons for UTSW, n=25 for Rockefeller), two pupils which share the same tutor (aka pupil vs. 'Sibling' comparisons, n = 60 comparisons for UTSW, n = 64 for Rockefeller), and between two pupils who don't share song tutor (aka pupil vs. unrelated bird, n = 90 comparisons for UTSW, n = 75 for Rockefeller). Calculated with a dataset of 30 typical tutor-pupil pairs from UTSW and 25 from Rockefeller. **c.** Correlation between EMD dissimilarity scores and human expert judgements of song similarity for 14 tutor-pupil comparisons from the UTSW colony (r = -0.87, p<0.005). **d.** Tutor-pupil EMD dissimilarity scores for typical pupils from the UTSW colony (n = 30), typical pupils from the Rockefeller Song Library (n = 25), and FP1 KD pupils from the UTSW colony (n = 7) (One Way ANOVA F(2, 57) = 18.6, p < 0.005. * indicates Tukey HSD post hoc p-adj < 0.05). **e.** EMD Dissimilarity score between birds at various age points across development, compared to their mature song recorded when the bird is over 90dph. Each point represents one comparison (n = 91 comparisons across 11 birds). Grey line is an exponential function fit to the data to emphasize the slowing of song maturation as birds approach maturity.

Using this method, we find that FP1 KD pupils have significantly higher EMD dissimilarity to tutor scores when compared to typical birds from either UTSW or Rockefeller (Fig 6d, One Way ANOVA F(2, 57) = 18.6, Tukey HSD FP1 KD vs typical UTSW p-adj < 0.005, FP1 KD vs typical Rockefeller p-adj < 0.005), showing that this method can be used to assess song learning outcomes in experimentally manipulated birds. We also used the model to look at how a bird's song changes over development, by comparing song at multiple age points to a bird's mature song. As expected, we find that birds gradually become more similar to their mature song over the course of development, and that the rate of this change slows as birds approach maturity (Fig 6e). Altogether, these tests showcase that this method is more reliable at assessing tutor-pupil song similarity than existing methods, while also not requiring any manual motif identification or dataset-specific fine tuning. As a result, as with the AVN acoustic, timing, and syntax features, its scores are directly comparable across research groups, facilitating the quantitative comparison of song learning outcomes across studies.

Discussion

Here, we have presented the AVN song analysis pipeline, which performs highly accurate syllable segmentation and syllable labeling. We have shown that this approach yields consistently high performance across multiple zebra finch colonies, suggesting that it can standardize and simplify large scale behavioral annotation across research groups, without the need for additional training or fine-tuning. The AVN labels are used to calculate syntax features which agree well with manual annotations, and which are sufficient to discriminate between typical birds and birds with known genetic disruptions. The AVN segmentations and raw song files are used to calculate timing features, which again are consistent across colonies, and which reflect a bird's stage in song development. Standard acoustic features are also calculated for each AVN syllable type, which can be used to describe the overall acoustic properties of a bird's song.

To showcase the utility of these song features, we presented how they can be used to compare multiple different song phenotypes, to test our hypothesis that the songs of FP1 KD birds would more closely

resemble isolate birds' compared to typical or deaf birds' songs [21]. We also showed how these features can be used to create an interpretable model to predict a bird's age within 7 days while their song is rapidly evolving from immature subsong to stable adult song. As more research groups use the AVN feature set to describe their birds' song phenotypes, these analyses will only become more sensitive and powerful. Ultimately, we hope that these song features can be used to establish a comprehensive map of song phenotypes, which more closely link abnormal song phenotypes with the neural circuit dysfunctions underlying them.

Finally, we developed a novel similarity scoring system which outperforms existing methods in its sensitivity and fidelity to expert human judgements of song similarity, all without requiring any manual song annotation. Again, we expect this to be an invaluable tool for describing the nature and severity of song learning phenotypes in experimentally manipulated birds, where existing similarity scoring methods perform particularly poorly.

AVN is available to researchers as an open-source python package and as a graphical application. The python package allows researchers with some coding experience to take full advantage of the flexibility of these tools and integrate this pipeline into their data collection and processing workflows, while the application allows other researchers to easily annotate their songs and calculate AVN features with minimal coding, in a highly reproducible fashion.

Altogether, we see this pipeline as an example of the integration of deep learning tools and expertly curated features to automated behavior analysis without compromising the interpretability or generalizability of results. This feature set and annotation approach was designed with zebra finches in mind, but should be easily adaptable to other species with discrete syllables that can be clustered according to their acoustic features, such as Bengalese finches and Canaries, for example [10]. These species have more complex syllable sequencing than zebra finches and would therefore also benefit from additional syntax and timing features specific to their species. Additionally, while we've strived for a comprehensive set of features, it is possible that our 55-feature set will fail to reflect certain interesting song phenotypes that haven't yet been observed. We hope that the open-source nature and extensive documentation of the AVN pipeline will allow and encourage researchers to contribute additional song features to the pipeline as they encounter such cases where the current feature set may be insufficient.

Methods

The AVN documentation, AVN-GUI, and code necessary to produce all figures in this manuscript can be found through the following links:

AVN Documentation: https://avn.readthedocs.io/en/latest/index.html

AVN GUI: https://avn.readthedocs.io/en/latest/AVN_GUI.html

Code for Figures: https://github.com/theresekoch/AVN_paper

Data Acquisition

A complete list of birds and the analyses in which they were included can be found in Supplementary Table 1.

UTSW Dataset

Many birds included in this study were previously recorded and analyzed in [21]. This includes 7 birds which were injected with a pscAAV-GFP-shFoxP1 virus before exposure to a song tutor, leading to disrupted songs (referred to as FP1 KD birds in this manuscript and FP1-KD SE in [21]). 8 birds were included in this group in the previous paper. One was omitted from this manuscript because it exhibited completely typical song, likely due to weak viral expression. A further 10 birds which were injected with a control virus before exposure to a song tutor (Ctrl SE in [21]), and 10 which were injected with the pscAAV-GFP-shFoxP1 after tutor song exposure (FP1-KD BI in [21]) are included in the current study. Both of these groups exhibit species-typical song production and are included in the 'typical' group in this study. Finally, 8 additional birds which were raised in isolation from an adult song model until they were at least 90 days post hatch ('Full isolates' in the FP1 paper and 'Isolates' in the current study) were included in this study, for a total of 35 birds. See [21] for more information on viral injections and rearing conditions.

An additional 8 adult isolate birds, 37 typically reared adult birds, and 10 juvenile birds were recorded for this study. For isolate birds, fathers were removed from breeding cages before the young reached 12dph. These young remained housed with their mother and siblings in a room containing only other isolate breeding cages, until they were weaned between 40 and 60dph, at which point they were housed individually, with auditory but no visual access to other isolate reared males. Typically reared and juvenile birds were raised in our main colony room which contained about 55 breeding pairs. They had unlimited access to their father's song in their home cages, until they were weaned between 40 and 60dph, at which point they were housed in group cages with other males.

Recordings were obtained by individually housing birds in sound attenuating chambers. They were continuously recorded using Sound Analysis Pro 2011 [30]. All birds were placed on a 14h:10h day:night

cycle and provided ad libitum access to food, water and grit. All procedures were performed in accordance with protocols approved by the Animal Care and Use Committee at UT Southwestern Medical Center.

Additional Song Data

In addition to the birds recorded at UTSW, this study includes recordings of 25 pupils and 6 tutors from the Rockefeller University Field Research Center Song Library [17], 5 juvenile birds from Duke University [9], and 5 early deafened and 4 sham deafened birds from Hokkaido University [31]. See citations for more information on rearing and recording conditions.

Manual Song Annotation

A random subset of 30 song files from a single day of recording were annotated for each of 35 adult birds from UTSW, and 15 song files were annotated for each of 25 adult birds from the Rockefeller Song Library. Manual annotation was performed using the *evsonganaly* application in MATLAB [35], and involved 1) amplitude threshold syllable segmentation with a threshold selected for each song file based on visual inspection of the amplitude trace and spectrogram, 2) manual correction of erroneous syllable onsets or offsets, and 3) assignment of syllable labels to each syllable based on visual inspection of the spectrogram.

For all applications except training TweetyNet [5], segments that reflect cage noise were dropped from the annotations based on visual inspection of the spectrograms. A second set of annotations were made which retained noise segments, labeling them as such, with all syllables and calls labeled as simply ‘vocalizations’ for the purpose of training TweetyNet.

Segmentation

Amplitude Segmentation

Amplitude segmentation was performed using the ‘RMSEDerivative’ class in the AVN python package’s segmentation module. Each song file is bandpass filtered between 200 Hz and 9000 Hz, then the root mean square energy (RMSE) of each audio frame is computed with a hop length of 512 samples, and a frame length of 2048 samples. The RMSE values of each song file are normalized, then the RMSE’s first derivative is compared against user-specified thresholds. A syllable onset is identified as a positive crossing of the ‘onset’ threshold. Syllable offsets tend to be marked by more gradual changes in RMSE compared to syllable onsets, making it difficult to identify them consistently. To mitigate this, we perform onset to onset segmentation with this method, meaning each segment included a song syllable, and the silent gap that immediately followed it. If a syllable onset is not followed by another onset within 300ms (as in the end of a song bout), the offset is set as the first negative crossing of an ‘offset’ threshold after the syllable onset.

In keeping with the TweetyNet and WhisperSeg segmentation methods which don’t require per-bird parameter adjustments, the same ‘onset’ and ‘offset’ thresholds were used for all birds in the dataset. These

thresholds were selected using AVN's segmentation.Utils.threshold_optimization_many_birds() function, which compares the F1 scores relative to manual segmentation obtained with multiple different threshold values to identify the threshold value that results in the lowest mean F1 score across all 35 UTSW birds used for amplitude segmentation validation. The same thresholds selected based on the UTSW birds were used to segment the 25 Rockefeller Song Library birds, as a test of the generalization of this method without the need for manual segmentations.

TweetyNet

The *vak* python package was used to prepare datasets for, train, and generate segmentation predictions with the TweetyNet model [5]. TweetyNet is a deep neural network consisting of a block of convolutional layers followed by a bidirectional long short-term memory (LSTM) layer. The model takes a 1s spectrogram of song as input, and labels each frame within that spectrogram. TweetyNet was designed for simultaneous syllable labeling and segmentation, in which case it would label each frame of the spectrogram with a syllable label or as silence. However, to make this model generalize to new birds without any additional training data, we instead trained TweetyNet to label each frame as a vocalization, silence, or noise (common sources of noise include the bird hopping around its cage and flapping its wings), rather than a more specific syllable type. When trained with such data from many birds, it can learn to distinguish vocalizations from noise and silence in a sufficiently general manner that the model can be applied to previously unseen individuals.

Manually annotated song files with label classes 'noise' and 'vocalization' were used to train TweetyNet in a leave-one-out cross validation scheme, meaning the model was trained with data from all but one bird and tested on the withheld bird for each of the 35 birds in the UTSW dataset. A model trained with data from all 35 UTSW birds was used to segment the 25 validation birds from the Rockefeller Song Library dataset, to test the model's ability to generalize to new colonies. Full model training and prediction procedures can be found in this paper's accompanying github repository. For more information on the TweetyNet model itself, see [5].

WhisperSeg

WhisperSeg is an instance of the Whisper Transformer model which was pre-trained for automatic human speech recognition, and fine-tuned for animal voice activity detection with a multi-species animal vocalization dataset [6]. It takes a spectrogram representation of up to 2.5s of song as input and outputs the indices of vocalization onsets and offsets in the spectrogram. These indices are then converted to timestamps, and a consistent labeling scheme for an entire song file is achieved through a 'majority-vote' post-processing step across overlapping 2.5s song segments. Syllable segmentation was performed using the *whisperseg-large-ms-ct2* model, with hyperparameters optimized for zebra finch song segmentation,

based on [6]. Full model prediction procedures can be found in this paper's accompanying github repository. For more information on the WhisperSeg model and its training, see [6].

Validation

Segmentation methods were compared on the basis of their precision, recall, and F1 scores relative to manual annotations.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{F1} = \frac{\text{True Positives}}{\text{True Positives} + \frac{1}{2}(\text{False Positives} + \text{False Negatives})}$$

where a true positive is a syllable onset in the automatic segmentation that is within 10ms of a syllable onset in the manual annotation, a false positive is a syllable onset in the automatic segmentation that doesn't match with a syllable onset in the manual annotation within 10ms, and a false negative is a syllable onset that is present in the manual annotation which doesn't match with an automatic segmentation onset within 10ms. When determining onset alignments, we ensure that each syllable onset in the manual annotation can only 'match' with a single syllable onset in the automatic segmentation, and vis-versa. This was done using AVN's `segmentation.Metrics.calc_F1()` function. Across all 3 metrics, scores closer to 1 indicate better agreement between automatic and manual segmentations. These same features were calculated for syllable offsets as well, but with an allowance of 20ms rather than 10ms, to account for the greater variability in exact offset segmentation across all methods tested.

To further examine the temporal precision of each method relative to manual annotation, we also calculated the time difference in milliseconds between matched syllable onsets and offsets between automatic segmentations and manual annotations. This was done using AVN's `segmentation.Metrics.get_time_delta_df()` function.

Labeling

UMAP Dimensionality Reduction

UMAP dimensionality reduction [18] is performed on spectrograms of syllables, prior to HDBSCAN clustering for label assignment [19]. First, spectrograms of each segmented syllable are produced. The audio is first bandpass filtered between 500Hz and 15KHz, then amplitude normalized independently for each syllable rendition. The short term Fourier transform of the normalized audio for each syllable is computed with a window length of 512 samples and a hop length of 128 samples. The resulting amplitude spectrogram is converted to decibels using the librosa `amplitude_to_db()` function [36]. The db-scaled spectrogram is then padded to match the dimensions of the longest syllable in a given bird's dataset, or

clipped to 870ms if it exceeds that duration. This limit is very generous, and only ever applies to segmentation errors, but it is necessary to avoid memory issues during UMAP computation. Normalized spectrograms are then flattened from an array to a single long vector, and the vectors corresponding to each spectrogram are concatenated into an array separately for each bird. This spectrogram array is used to calculate the UMAP embeddings of each syllable using the UMAP python package's `UMAP()` function. This approach is based on [10].

At a high level, UMAP dimensionality reduction involves constructing a graphical representation of the syllable set, where each syllable spectrogram can be thought of as a point in high dimensional space which is connected to other syllables near it by edges. These edges are weighted based on the distance between points and the local density of those data points. The high dimensional graph is then projected into lower dimensions in a way that best preserves its overall structure.

UMAP dimensionality reduction can be a useful initial step when attempting to cluster high-dimensional data points because many clustering algorithms, especially density-based clustering algorithms such as HDBSCAN can suffer from the 'curse of dimensionality'. When clustering spectrograms directly, each pixel in the spectrogram is a dimension, meaning each spectrogram exists as a point in a space with thousands of dimensions. In such a high dimensional space, points will be very sparsely distributed, even if the spectrograms appear largely very similar. As a result, it is very difficult to detect regions of higher point density to serve as the basis of clusters. Reducing the dimensionality of the dataset forces points closer together, such that regions of high density separated by lower density can be more easily detected. UMAP is particularly adept at emphasizing local clusters in high dimensional data because of how its initial embedding graph is constructed.

UMAP parameters were selected based on suggestions in the UMAP-learn documentation for clustering using UMAP embeddings and based on visual inspection of plots and labeling outcomes compared to manual annotations for birds from the UTSW dataset.

HDBSCAN Clustering

The "Hierarchical Density-based Spatial Clustering of Applications with Noise" (HDBSCAN) clustering algorithm [19] was applied to the UMAP embeddings of syllable spectrograms for each bird independently, in order to assign syllable labels. This method was selected based on the results in [10] for clustering Bengalese finch syllables, a species closely related to zebra finches.

Essentially, HDBSCAN works by calculating the 'mutual reachability' distance between points in the UMAP space, based on the distance between them and their local densities. These mutual reachability distances serve as edges connecting nodes (points representing individual song syllables) in a graph, which

are then pruned to obtain a minimum spanning tree (a graph using the minimum number of total edges to connect all points). The minimum spanning tree is then converted to a hierarchy by sorting the edges on the basis of their mutual reachability scores. Clusters of points are identified by defining a minimum cluster size and selecting the clusters that persist over the longest span of the hierarchy. As with the UMAP parameters, the same HDBSCAN hyperparameter set was used for all birds. The hyperparameter values were selected based on v-measure scores and visual inspection of confusion matrices for WhisperSeg segments compared to manual annotations for birds from the UTSW colony.

Validation

Syllable labeling was assessed by comparing automatically assigned syllable labels to manual annotations. Automatically labeled segments first had to be aligned to the manual annotations to identify pairs of labels in the automatic clustering and manual annotation that referred to the same vocalization. This was done using the same method described in the *Segmentation Validation* section, in which syllable onsets are uniquely matched to their closest counterpart across segmentation methods, this time up to a maximum distance of 100ms. False positive syllable detections (i.e. syllable present in the automatic segmentation without a manual annotation counterpart) are assigned to their own manual annotation category ('x'), and False negative syllables detections (i.e. syllables present in the manual annotation without an automatic segmentation counterpart) are assigned to their own cluster ('1000') for the purposes of visualization and quantification.

Once syllables have been aligned between automatic segmentation and manual annotations, the HDBSCAN cluster labels are compared to manual labels for each bird to construct a confusion matrix, which gives the number of syllables in each HDBSCAN cluster that carry each of the possible manual labels. The confusion matrix values can then be used to compute homogeneity, completeness, and v-measure scores, to evaluate the correspondence between HDBSCAN labels and manual annotations for each bird. Homogeneity measures the extent to which syllables with the same AVN label also carry the same manual annotation label, completeness measures the extent to which syllables with the same manual annotation label also carry the same AVN label, and the V-measure is the harmonic mean of these two scores.

$$\text{Homogeneity} = \frac{\text{H}(\text{manual labels} \mid \text{clusters})}{\text{H}(\text{manual labels})}$$

$$\text{Completeness} = \frac{\text{H}(\text{clusters} \mid \text{manual labels})}{\text{H}(\text{clusters})}$$

$$\text{V-measure} = \frac{2 \times \text{Homogeneity} \times \text{Completeness}}{\text{Homogeneity} + \text{Completeness}}$$

Where $H(\text{manual labels} \mid \text{clusters})$ is the conditional entropy of the manual labels given the cluster labels, $H(\text{manual labels})$ is the entropy of the manual labels, and vis-versa. In all cases, a higher score indicates

better correspondence between clusters and manual labels, with a maximum possible score of 1, and minimum score of 0.

Syntax Features

Syntax Raster Plot

Beginning with a table of all AVN labels, syllables that are preceded and followed by a period of silence longer than 200 ms are removed, as they likely reflect calls produced outside of song. Song bouts are then identified as sequences of at least two syllables that are separated by silent gaps no longer than 200ms. These bouts are aligned based on a user-specified alignment syllable, such that the first instance of the alignment syllable is in the same position across all bouts. This alignment is important as bouts typically begin with a variable number of introductory notes, which will obscure patterns in syllable sequence across bouts when they are not aligned to the first non-introductory note syllable. After alignment, bouts are ordered such that bouts with similar sequences after the alignment syllable are together in the final plot, which also helps emphasize patterns across bouts. This is done using AVN's syntax.Syntax_Data.make_syntax_raster() function. See avn's documentation for additional information and examples.

Syllable Transition Matrix

As with syntax raster plots, syllables that are preceded and followed by more than 200ms of silence are dropped from the AVN labels as they likely reflect calls produced outside of song. Silent gaps longer than 200ms and file bounds are then added as states to the AVN label sequence. All syllable transitions between AVN labels are then counted, including transitions to and from periods of over 200ms of silence; meaningful transitions as they reflect the beginnings or ends of song bouts. Transitions to and from file bounds are ignored, as these are artifacts of the recording and don't reflect meaningful behavioral states. The transition counts are then divided by the total number of renditions of the first syllable type in the transition to get the conditional probability of the second syllable, given the first syllable. This is done using AVN's syntax.Syntax_Data.make_transition_matrix() function.

Syntax Entropy Rate

Syntax stereotypy is quantified using the entropy rate of the syllable transition matrix.

$$\text{entropy rate} = -\sum_{\text{i, k}} \pi_i p_{i,k} \log_2(p_{i,k})$$

where $p_{i,k}$ is the probability of transitioning from initial syllable type *i* to following syllable type *k*, and π_i the probability of syllable type *i* occurring, regardless of what syllable precedes or follows it. An entropy rate approaching 0 indicates that all transitions are highly predictable. The maximum possible entropy rate score is $log_2(N)$ where *N* is the number of syllable types in the bird's repertoire plus one to account for silence as a possible state. To directly compare scores between birds without being biased by the number of syllable

types in their song (a feature which depends strongly on the number of syllable types present in their tutor's song), we divide the entropy rate score by $log_2(N)$ such that is it now bounded between 0 and 1. This is done using AVN's syntax.Syntax_Data.get_entropy_rate() function.

Repetition Bouts

A repetition bout refers to every instance in which a syllable is produced, either a single time or multiple times in a row. For example, in the syllable sequence *abcaaabc*, syllable *a* has 2 repetition bouts, one of length one, meaning the syllable was produced without being repeated, and one of length 3, meaning the syllable was produced 3 times in a row. The number and length of repetition bouts is calculated for each syllable type in a bird's repertoire. The mean repetition bout length and coefficient of variation (CV) of repetition bout length is then calculated for each syllable type.

To facilitate comparisons across birds, which have different numbers of syllable types, the mean repetition bout length and CV of repetition bout length for the syllable type with the highest mean repetition bout length are selected to represent the bird's overall tendency to repeat syllables, excluding syllable types that reflect putative calls or introductory notes. Typical zebra finches often repeat calls or introductory notes, but rarely repeat song syllables, so looking at the repetition of song syllables is more informative when detecting or comparing birds with abnormal syntax. That said, in certain experiments repetition bout features of calls or introductory notes may be of greater interest, in which case they can also be specifically identified using AVN.

An AVN syllable type is considered a putative introductory note if it 1) is no less than 5% less likely to be transitioned to from silence than the syllable type most commonly transitioned to from silence, meaning it tends to occur at the start of a vocalization bout, and 2) it has a single dominant transition to a syllable type other than itself which is not silence, meaning that after a number of repetitions, it is eventually followed by a predictable next syllable type, which should reflect the start of a motif. These criteria were determined based on inspection of the syntax properties of introductory notes in manual song annotations. An AVN syllable type is considered a putative call if it is 1) not a putative introductory note, and 2) is produced in a bout of one or two syllables preceded and followed by at least 200ms of silence in more than ⅓ of all utterances. This criterion was again determined based on visual inspection of manual song annotations.

Song Timing Features

Syllable and Gap Duration Entropy

A syllable duration distribution is constructed based on the segment durations output by WhisperSeg for each bird. A histogram of the $\log_{10}$ of syllable durations is calculated, with 50 evenly spaced bins ranging from -2.5 to 0. As in [28] the log of syllable durations are used because the syllable duration distributions of juvenile birds are roughly exponential, and therefore linear in log space. Histograms are normalized to

produce a probability density function across syllable durations. The entropy of this distribution is then calculated as

$$\text{Entropy} = \frac{\sum_{i=1}^{N} p_i \log(p_i)}{\log(N)}$$

where p_i is the density the *i*th bin in the histogram, and *N* is the total number of bins (50, in this case). The resulting entropy can range from 0 to 1, with higher scores indicating less predictable syllable durations, consistent with the songs of immature birds.

Entropy is calculated similarly for silent gap durations, where gaps durations are defined as the time difference between a syllable offset and the immediately following syllable onset, up to a maximum duration of 200 ms. A log transform was not applied to gap durations before constructing a histogram with 20 10ms bins.

Rhythm Spectrograms

Rhythm spectrograms are a visualization of the strength and stereotypy of rhythmic patterns in a bird's song, generated by concatenating the rhythm spectra of multiple song bouts, as first proposed in [29]. A song bout's rhythm spectrum is the spectrum of the first derivative of its amplitude. If a song's amplitude has consistent repeating fluctuation patterns (as we expect for a bout composed of multiple repetitions of the same stereotyped motif), then its spectrum will exhibit harmonic banding patterns. If, by contrast, there are no repeating rhythms in the song's amplitude, the rhythm spectrum will have a more even spread of energy across frequency bands. To detect these harmonic patterns more easily in the rhythm spectrogram, all rhythm spectrograms are plotted as the rolling average of 10 song bouts, smoothing out some bout-to-bout variation in the spectra to make harmonic bands more obvious.

To ensure consistent dimensions and resolution across song bouts, the rhythm spectrum is actually calculated for segments of song of a fixed duration, rather than complete song bouts. This also eliminates the need for any segmentation and labeling of song files to identify bouts, making this timing analysis method completely independent of possible segmentation and labeling errors. Each .wav file is broken into multiple 3-second-long frames, with a hop length of 0.2 seconds. The 3 frames with the highest total amplitude (ie the 3 windows containing the most vocalizations) from each file have their rhythm spectra calculated, and the mean of their spectra is taken as the rhythm spectrum for that file. Because of this windowing system, only files at least $3 + 3 \times 0.2$ seconds in duration can be windowed this way, so shorter .wav files are ignored.

The derivative of the amplitude of each frame is centered at 0 by subtracting the mean value, then multiplied by a Hanning window to reduce spectral leakage when calculating the spectrum. The transformed

amplitude derivative is then padded to a total length of 100000 frames, resulting in a smoother spectrum with more interpolated values. A bandpass filter is then applied, keeping only frequency components above 1 Hz and below 500Hz, as these are the frequencies consistent with typical zebra finch motif and syllable periods. Finally, the real component of the Fourier transform is calculated, constituting the frame's 'rhythm spectrum'. Only portions of the rhythm spectrum corresponding to frequencies between 0 and 30Hz are included in rhythm spectrograms and downstream feature calculations, as this is the range with the strongest harmonic banding for typical zebra finches. This is all done using AVN's avn.timing.RhythmAnlysis.make_rhythm_spectrogram() function.

Rhythm Spectrum Entropy

We quantify the strength of the harmonic content of a bird's rhythm spectrum (i.e. the strength of its rhythm) by calculating the Wiener Entropy of the mean rhythm spectrum across bouts. Wiener entropy is a common acoustic feature used to assess the harmonic nature of zebra finch syllables, with scores near 0 reflecting signals with little harmonic structure, and scores ranging to negative infinity for signals with more harmonic structure.

$$\text{rhythm spectrum entropy} = \frac{\sum_{n=0}^{N} \log(\text{rhythm spectrum}^2)}{N - \log(\sum_{n=0}^{N} \text{rhythm spectrum}^2)/N)}$$

This is calculated using AVN's avn.timing.RhythmAnalysis.calc_rhythm_spectrogram_entropy() function.

Peak Frequency Variability

Whereas the rhythm spectrum entropy measures the overall strength of the rhythms in a set of songs, the peak frequency variability reflects the consistency of the rhythm across multiple song renditions. The exact spacing of the harmonics in a rhythm spectrogram depends on the shape of the amplitude trace of the bird's motif. It isn't obvious how different motifs and motif lengths affect the banding pattern of the rhythm spectrogram, so it doesn't make sense to compare the appearance of different birds' rhythm spectra beyond the prominence of harmonic bands. Likewise, the frequency of the harmonic band with the highest magnitude doesn't carry any special meaning. However, in a very stereotyped bird, that harmonic band will be consistent across songs. If the bird sings its song slightly faster or slower the band can shift slightly in the frequency domain. So, we measure a bird's rhythm stereotypy by looking at the variability of the frequency with the highest magnitude across song files (the peak frequency).

In practice the frequency band with the highest energy can jump between harmonic bands across files, even while the overall timing is largely unchanged, so to truly capture fluctuations in the underlying rhythms in a bird's song, we restrict the range of 'peak frequency' values to a 3Hz band about the median peak frequency across bouts. The coefficient of variation of the peak frequency within this range is calculated

as the peak frequency variability. This is done using AVN's avn.timing.RhythmAnalysis.calc_peak_freq_cv() function.

Acoustic Features

Goodness of pitch, Mean Frequency, Weiner Entropy, Amplitude, Amplitude Modulation, Frequency Modulation, and Pitch were all calculated using AVN in python, with implementations based on the Sound Analysis Tools for MATLAB [30]. Each of these features is calculated for each frame in a spectrogram, resulting in a time series of values. We summarize these time series of varying lengths by taking the mean value of each feature for each AVN segmented syllable. We then calculate the mean and coefficient of variation of the mean feature values for each syllable type according to their AVN labels. As each bird has a different number of syllable types, we need to further summarize these features so that we have a consistent set of values for comparisons across individuals. To do this, we take the syllable type with the minimum, maximum and median mean value and CV for each feature. This results in 6 values summarizing the variability and absolute values of each feature for each bird. Across 7 acoustic features plus syllable duration, this results in a total set of 48 features.

Linear Discriminant Analysis

We fit 3 different linear discriminant analysis models in this paper. One to discriminate between typical zebra finches and isolate zebra finches, one to discriminate between typical zebra finches and deaf zebra finches, and one to discriminate between all 3 groups at once. For each of these models, L1 regularization was used to reduce the number of features considered in the model. This improves both the generalization of the model, and its interpretability by focusing on just a subset of the most informative features. L1 feature selection was performed considering all AVN features from each bird, excluding amplitude and amplitude-modulation features, as these were found to vary according to recording conditions. Once the feature set was reduced, classification accuracy of the models was tested using a stratified k-folds cross validation approach. Plotted LDA values and feature weights were obtained from a model trained with the complete dataset. This was all done using the scikit-learn python package [37].

Age Prediction Generalized Additive Model

The full AVN feature set was calculated for 19 individual birds across 103 age points, using songs produced within the first 4 hours after lights on. Juvenile birds have been shown to have more variable songs in the early morning [9], which exaggerates the difference between immature and mature song and improves the model's ability to predict a bird's age, compared to features calculated with a full day of songs, or songs produced in the afternoon.

Before fitting a Generalized Additive Model (GAM) for age prediction, we pruned our feature set to include only the most informative features. We first excluded all amplitude and amplitude-modulation

features, as these were strongly affected by recording conditions and differed between colonies. We then calculated the mutual information between each remaining feature and age, considering 43 age points from 12 individual birds. We automatically excluded all features with a mutual information score lower than 0.05 (20/44 features). We further refined this feature set by performing forward feature selection with our 43 age point dataset. This means we iteratively added individual features to the model based on which additional feature resulted in the lowest mean squared error (MSE) predictions in a bird-fold cross validation. We then selected the feature set with the lowest overall MSE, further reducing our feature set to just 7 features.

A GAM model with the 7 selected features was used to predict bird's ages in a leave-one-bird-out (aka bird-fold) cross validation scheme. Here, we included the 43 age points from 12 individual birds used for feature selection, plus an additional test set of 60 age points from 7 individual birds. We saw no significant difference in model performance between this test set and the dataset used for feature selection, so we pooled results across these groups.

To investigate the contribution of each feature to the overall model, we fit a model with all birds in the dataset, and used the pyGAM python package [38] to extract the partial dependence functions for each feature.

Similarity Scoring

Data Preparation

Spectrograms of manually segmented and labeled syllables from 21 adult zebra finches from the UTSW colony were used for model training. All validation was performed with spectrograms of WhisperSeg segmented syllables from a test set of 30 tutor-pupil pairs from UTSW and 25 tutor pupil pairs from the Rockefeller Song Library [17], none of which were included in training. These spectrograms are normalized for amplitude, then clipped or padded to a uniform duration of 180ms. To reduce computational costs, all frequency bands below 2kHz and above 6kHz are discarded.

Model Architecture

The proposed neural network model is composed of 5 convolutional layers alternating with 4 'Multiscale Analysis Modules' (MAMs), followed by a global pooling layer and 3 fully connected linear layers (supplemental Fig11). This architecture is based on the model proposed in [39], which was used for species classification with field recordings of bird, frog and toad vocalizations. The first convolutional layer consists of 32 3 x 3 kernels, with the 4 subsequent convolutional layers consisting of 64 3 x 3 kernels with a stride length of 2 along the frequency axis, resulting in down sampling by a factor of 2 along that dimension. Each MAM is composed of 4 parallel strands, each processing the data at different scales. The first strand consists of a single convolutional layer with 32 1 x 1 kernels, the second, third and fourth strands start with a 32 filter 1 x 1 kernel convolutional layer, followed by a convolutional layer with 32 3x3, 5x5, and 7x7 kernels,

respectively. The output of each of these strands is concatenated channel-wise, resulting in a 128 channel representation of the data which is passed to the next layer. This parallel strand organization allows the model to perform feature extraction at multiple different scales without increasing the depth of the model, saving computational cost and limiting potential overfitting. This approach was first proposed in [40]. The ReLU activation function is used after every layer [41]. The output of the final linear layer is an 8-dimensional vector, which represents an input syllable's embedding. These vectors are normalized to have a length of 1, such that all embeddings lie on a unit 8-dimensional hypersphere.

Model Training

The model is trained using dynamic triplet loss with triplet mining. Triplet loss involves presenting the model with batches of triplets, where each triplet consists of an anchor, a positive, and negative syllable. The anchor and positive syllables carry the same manual annotation label from the same bird, and the anchor and negative syllables carry different labels, either from the same bird or from an unrelated bird. The loss function to be minimized is:

$$\text{Loss} = \sum_{i=1}^{N} \max\left(\left|\left|f(A_i) - f(P_i)\right|\right|_2 - \left|\left|f(A_i) - f(N_i)\right|\right|_2 + \alpha,\ 0\right)$$

where *N* is the total number of possible triplets in training, $f(A_i)$ is the embedding of the anchor in triplet *i*, $f(P_i)$ is the embedding of the positive, $f(N_i)$ is the embedding of the negative, and α is a margin parameter. If the positive is closer to the anchor than the negative by at least α, the loss for that triplet is 0.

During training, many randomly sampled triplets will already yield a loss of 0, and therefore will not lead to any change in the model. As a result, such triplets are not presented to the model during training. The remaining triplets which result in a positive loss value can be divided into two groups, hard triplets and semi-hard triplets. Hard triplets are cases where $||f(A) - f(P)|| > ||f(A) - f(N)||$, and semi-hard triplets are cases where $||f(A) - f(P)|| < ||f(A) - f(N)||$ and $||f(A) - f(P)|| - ||f(A) - f(N)|| > \alpha$. Hard triplets result in higher loss values, and therefore larger model weight updates, which can result in unstable training. Previous studies have shown that, as a result, training with semi-hard triplets alone can lead to faster and more stable model convergence [42]. We found that we achieved the best model performance when training with a ratio of 75 semi-hard triplets : 25 hard triplets, so this ratio was used to train the final model. A forward pass of the model is performed for each mini batch in training to determine the 'hardness' of all possible triplets from that batch. Triplets are then sampled according to the specified ratio of semi-hard to hard triplets, and are presented to the model for training.

Our approach is said to be *dynamic* triplet loss because the value of the margin parameter α is updated dynamically over the course of training. The value is initially set to 0.1 and increased stepwise by

0.2 every time a training epoch had fewer than 2,500 hard or semi hard triplets per batch on average, up to a maximum value of 0.7. This allows the model to begin learning an easier task, of separating syllables of different classes by a smaller margin. As the margin increases, the task gradually becomes more difficult, leading to more stable model convergence compared to starting with a higher margin value. Each time the margin parameter is increased, triplets that previously resulted in a loss of 0 can become semi-hard triplets, meaning the set of triplets presented to the model also expands over the course of training. The initial margin value, maximum margin value, margin step size and non-zero triplet threshold were all determined empirically. The weight optimization was performed with the Adam optimizer, with weight decay of 0.0001 [43].

Ultimately, this dynamic triplet loss training constitutes a form of deep metric learning, where high dimensional inputs (e.g., spectrograms of syllables) are mapped onto a lower-dimensional space where the similarity between samples is proportional to the distance between them. Training with triplets is advantageous in a context where the total amount of labeled training data is limited, as the number of possible triplets is proportional to the cube of the number of training samples.

EMD

Syllable embeddings were obtained by running a forward pass of the trained model with all segmented syllables that an individual bird produced on a given day. This results in a distribution of thousands of syllables in the 8-dimensional embedding space. Two birds songs are compared by calculating the Earth Mover's Distance (EMD) between their syllable embedding distributions using the PyEMD package [44]. If one were to imagine the distributions as piles of dirt, the earth mover's distance is the minimum cost of moving earth from one distribution to match the other, where cost is defined as the amount of dirt moved multiplied by the distance over which it is moved. This value can range from 0 for identical distributions, to positive infinity for distributions that are infinitely far apart. As our embedding space is limited to points lying on a unit radius 8-dimensional hypersphere, the maximum possible value for EMD is 1.41 for distributions that are each concentrated on a single point, maximally separated within the constraints of the embedding space.

The EMD score considers many song renditions from each bird being compared, allowing a better overall comparison of the similarity between two birds' song production as compared to multiple pairwise comparisons between renditions. One limitation of EMD, however, is that it is completely agnostic to syllable sequencing, so a pupil that imitated all syllables from his tutor but sings them in a completely different order will have a similar EMD score to a pupil that imitated all syllables from a tutor and produces them in the same order. That said, there is new evidence that zebra finches recognize songs independently of syllable order, raising questions about the importance of syllable order in song perception [45]. The EMD score is also symmetrical, meaning that the presence of syllables in the tutor's song that weren't imitated by the pupil

will have the same impact on the EMD score as new, improvised syllables present in the pupil's song but not in the tutor's song.

Similarity Scores across Comparison Types

To validate the performance of our model and EMD scores for similarity scoring, we compute EMD scores between a pupil and itself, a pupil and its tutor, pupil and a 'sibling', and a pupil and an unrelated bird. For comparisons between a pupil and itself, embeddings of all recorded syllables from a day of song are computed. If the bird has fewer than 4,000 syllables in this dataset, the dataset is randomly split in half and the two halves are compared. If a bird has more than 4,000 syllables, two sets of 2,000 syllables are randomly sampled and compared. For comparisons between a pupil and its tutor, up to 4,000 syllables are sampled from each of the tutor and the pupil and compared. In the case of pupil and 'sibling' comparisons, a sibling is defined as another bird sharing the same song tutor. We expect that typical zebra finches that learned from the same tutor will have similar songs, but that these will generally be less similar than a pupil compared to its tutor directly. Each pupil is compared to up to 3 'siblings', depending on availability in our dataset. Finally, each pupil is compared to 3 randomly selected pupils who don't share their song tutor. For each of these comparisons, up to 4,000 syllables are randomly sampled from each bird as well, to help reduce the compute time for EMD.

Contrast Index

As in [32], contrast index is calculated as:

$$\text{Contrast Index} = \frac{\text{self similarity - cross similarity}}{\text{self similarity + cross similarity}}$$

where *self similarity* is the EMD score between pupil and itself, calculated as described in the previous section, and *cross similarity* is the mean similarity between a pupil and 3 unrelated birds, again calculated as described in the previous section. As EMD is a *dissimilarity* score, rather than a similarity score, more negative values actually reflect better contrast between comparisons, so we report the absolute value for ease of comparison to existing similarity scoring methods.

Expert Human Similarity Scores

A panel of 11 expert human annotators were each presented with 126 pairs of spectrograms and were instructed to rate their similarity on a scale from 1 (not similar) to 10 (very similar). The spectrograms were generated using Sound Analysis Pro 2011 [30], began at the beginning of a song bout, and included at least one full motif when motif structure was present. Raters were presented with 4 pupil-tutor spectrogram pairs per pupil, 8 comparisons between a tutor and itself to ensure that raters were using the full rating scale, and 10 duplicated tutor-pupil comparisons to ensure that the scorers were internally consistent. No individual scorer differed from the mean score by more than an average of 2 standard deviations, none differed by more than 2 points on the duplicated comparisons, and all but one made use

of the full scale (this scorer never assigned a perfect 10/10 score, so their scores were rescaled such that they spanned the full range). Similarity scores for each tutor-pupil pair were obtained by taking the mean similarity score across their 4 spectrogram pairs, across all scorers. The full scoring set is available at https://forms.gle/9TDu1fwGGYXWKhgB6. These scores were previously generated for and published in [21]. Of the birds evaluated, 15 were also in the similarity scoring validation set, so the correlation between these 15 birds' mean human similarity scores and tutor-pupil EMD scores were used to evaluate the agreement between methods.

For comparison to the expert human similarity scores, Sound Analysis Pro 2011 % similarity scores were calculated for the same set of 15 pupils. A representative motif from the tutor song was selected and compared to between 30 and 60 motif renditions from the pupil bird when the pupil was over 90dph, using the asymmetric time-courses similarity tool. The final reported scores are the mean %similarity across all comparisons for a given pupil.

Comparisons to Mature Song

For the 6 birds from UTSW and 5 birds from Duke University [9] from which we had recordings at 90-100dph and earlier time points, we computed the EMD between their juvenile and adult songs. For each bird, 4,000 WhisperSeg segmented syllables were sampled from a full day of song recordings when the birds were between 90 and 100 dph, to serve as the mature song distribution. Up to 4,000 WhisperSeg segmented syllables were sampled from each day of available recordings prior to or shortly following the mature song date for comparison. EMD scores were calculated using embeddings from the similarity scoring model as described previously. As the scores appeared to follow an exponential pattern, where the rate of song dissimilarity change slowed over development, we fit an exponential function to the data using the scipy.optimize curve_fit() function [46], and plotted this function alongside the data.

Data Availability

Song recordings and annotations for all birds recorded at UTSW are available through the Texas Data repository (https://dataverse.tdl.org/dataverse/avn). Annotations of songs from the Rockefeller University Song Library [17] generated for this paper are also available through the Texas Data Repository (https://doi.org/10.18738/T8/DN0SIV), while the songs are available at http://ofer.hunter.cuny.edu/songs. Recordings of juvenile birds from Duke University are available at https://doi.org/10.7924/r4j38x43h. Recordings of early-deafened zebra finches are available upon request from Dr. Kazuhiro Wada.

Acknowledgements

We would like to thank Fayha Zia for help with manual syllable labeling, Chihito Mori and Kazuhiro Wada for sharing recordings of early-deafened zebra finches, Ofer Tchernichovski and Erich Jarvis for publicly sharing the Rockefeller University Field Research Center Song Library, and Richard Mooney for publicly sharing recordings of juvenile zebra finches across development. We thank Tyler Lee and Daisuke Hattori for their valuable feedback and suggestions on AVN's design and validation. We also thank Luis Garcia, Andrea Guerrero, Jennifer Holdway and all members of the Roberts Lab for bird care, support, and their generous feedback on this work.

Funding This research was supported by the US National Institutes of Health R01 DC020333 to TFR. TMIK was supported by a Neural Scientist Training Program Fellowship from the UT Southwestern O'Donnell Brain Institute.

References

1. Wiltschko, A.B., et al., *Revealing the structure of pharmacobehavioral space through motion sequencing*. Nature Neuroscience, 2020. **23**(11): p. 1433-1443.
2. Hsu, A.I. and E.A. Yttri, *B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors*. Nature Communications, 2021. **12**(1): p. 5188.
3. Alam, D., F. Zia, and T.F. Roberts, *The hidden fitness of the male zebra finch courtship song*. Nature, 2024. **628**(8006): p. 117-121.
4. Steinfath, E., et al., *Fast and accurate annotation of acoustic signals with deep neural networks*. eLife, 2021. **10**: p. e68837.
5. Cohen, Y., et al., *Automated annotation of birdsong with a neural network that segments spectrograms*. eLife, 2022. **11**: p. e63853.
6. Gu, N., et al., *Positive Transfer of the Whisper Speech Transformer to Human and Animal Voice Activity Detection*. bioRxiv, 2023: p. 2023.09.30.560270.
7. Coffey, K.R., R.E. Marx, and J.F. Neumaier, *DeepSqueak: a deep learning-based system for detection and analysis of ultrasonic vocalizations*. Neuropsychopharmacology, 2019. **44**(5): p. 859-868.
8. Goffinet, J., et al., *Low-dimensional learned feature spaces quantify individual and group differences in vocal repertoires*. eLife, 2021. **10**: p. e67855.
9. Brudner, S., J. Pearson, and R. Mooney, *Generative models of birdsong learning link circadian fluctuations in song variability to changes in performance*. PLOS Computational Biology, 2023. **19**(5): p. e1011051.
10. Sainburg, T., M. Thielk, and T.Q. Gentner, *Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires*. PLOS Computational Biology, 2020. **16**(10): p. e1008228.
11. Roeser, A., et al., *The songbird lateral habenula projects to dopaminergic midbrain and is important for normal vocal development*. 2023, Cold Spring Harbor Laboratory.
12. Doupe, A.J. and P.K. Kuhl, *BIRDSONG AND HUMAN SPEECH: Common Themes and Mechanisms*. Annual Review of Neuroscience, 1999. **22**(Volume 22, 1999): p. 567-631.
13. Lachlan, R.F., et al., *Zebra Finch Song Phonology and Syntactical Structure across Populations and Continents-A Computational Comparison*. Front Psychol, 2016. **7**: p. 980.

14. Tchernichovski, O. and F. Nottebohm, *Social inhibition of song imitation among sibling male zebra finches*. . Proceedings of the National Academy of Sciences, 1998. **95**(15).
15. Scharff, C. and F. Nottebohm, *A comparative study of the behavioral deficits following lesions of various parts of the zebra finch song system: implications for vocal learning*. The Journal of Neuroscience, 1991. **11**(9): p. 2896-2913.
16. Koumura, T. and K. Okanoya, *Automatic Recognition of Element Classes and Boundaries in the Birdsong with Variable Sequences*. PLOS ONE, 2016. **11**(7): p. e0159188.
17. Tchernichovski, O., S. Eisenberg-Edidin, and E.D. Jarvis, *Balanced imitation sustains song culture in zebra finches*. Nature Communications, 2021. **12**(1): p. 2562.
18. McInnes, L., J. Healy, and J. Melville, *Umap: Uniform manifold approximation and projection for dimension reduction*. arXiv preprint arXiv:1802.03426, 2018.
19. McInnes, L. and J. Healy. *Accelerated hierarchical density based clustering*. in *2017 IEEE international conference on data mining workshops (ICDMW)*. 2017. IEEE.
20. Hyland Bruno, J. and O. Tchernichovski, *Regularities in zebra finch song beyond the repeated motif*. Behavioural Processes, 2019. **163**: p. 53-59.
21. Garcia-Oscos, F., et al., *Autism-linked gene FoxP1 selectively regulates the cultural transmission of learned vocalizations*. Science Advances, 2021. **7**(6): p. eabd2827.
22. Xiao, L., et al., *Expression of FoxP2 in the basal ganglia regulates vocal motor sequences in the adult songbird*. Nature Communications, 2021. **12**(1): p. 2617.
23. Tanaka, M., et al., *Focal expression of mutant huntingtin in the songbird basal ganglia disrupts cortico-basal ganglia networks and vocal sequences*. Proc Natl Acad Sci U S A, 2016. **113**(12): p. E1720-7.
24. Sánchez-Valpuesta, M., et al., *Corticobasal ganglia projecting neurons are required for juvenile vocal learning but not for adult vocal plasticity in songbirds*. Proc Natl Acad Sci U S A, 2019. **116**(45): p. 22833-22843.
25. Norton, P., et al., *Differential Song Deficits after Lentivirus-Mediated Knockdown of FoxP1, FoxP2, or FoxP4 in Area X of Juvenile Zebra Finches*. The Journal of Neuroscience, 2019. **39**(49): p. 9782-9796.
26. Kubikova, L., et al., *Basal ganglia function, stuttering, sequencing, and repair in adult songbirds*. Sci Rep, 2014. **4**: p. 6590.
27. Aronov, D., et al., *Two distinct modes of forebrain circuit dynamics underlie temporal patterning in the vocalizations of young songbirds*. J Neurosci, 2011. **31**(45): p. 16353-68.
28. Goldberg, J.H. and M.S. Fee, *Vocal babbling in songbirds requires the basal ganglia-recipient motor thalamus but not the basal ganglia*. J Neurophysiol, 2011. **105**(6): p. 2729-39.
29. Saar, S. and P.P. Mitra, *A technique for characterizing the development of rhythms in bird song*. PLoS One, 2008. **3**(1): p. e1461.
30. Tchernichovski, O., et al., *A procedure for an automated measurement of song similarity*. Animal Behaviour, 2000. **59**(6): p. 1167-1176.
31. Mori, C. and K. Wada, *Audition-Independent Vocal Crystallization Associated with Intrinsic Developmental Gene Expression Dynamics*. The Journal of Neuroscience, 2015. **35**(3): p. 878-889.
32. Mandelblat-Cerf, Y. and M.S. Fee, *An Automated Procedure for Evaluating Song Imitation*. PLOS ONE, 2014. **9**(5): p. e96484.
33. Lachlan, R.F., *Luscinia: a bioacoustics analysis computer program*. See luscinia. sourceforge. net.[Google Scholar], 2007.
34. Mets, D.G. and M.S. Brainard, *An automated approach to the quantitation of vocalizations and vocal learning in the songbird*. PLOS Computational Biology, 2018. **14**(8): p. e1006437.
35. Tumer, E.C. and M.S. Brainard, *Performance variability enables adaptive plasticity of 'crystallized' adult birdsong*. Nature, 2007. **450**(7173): p. 1240-1244.
36. McFee, B., Matt McVicar, Daniel Faronbi, Iran Roman, Matan Gover, Stefan Balke, Scott Seyfarth, Ayoub Malek, Colin Raffel, Vincent Lostanlen, Benjamin van Niekirk, Dana Lee, Frank Cwitkowitz, Frank Zalkow, Oriol Nieto, Dan Ellis, Jack Mason, Kyungyun Lee, Bea Steers, ... Waldir Pimenta., *librosa/librosa: 0.10.1*. 2023: Zenodo.
37. Fabian Pedregosa, G.V., Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David

Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay, *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 2011. **12**: p. 2825-2830.

38. Servén, D., C. Brummitt, and H. Abedi, *pyGAM*. 2018: Zenodo.
39. Thakur, A., et al., *Deep metric learning for bioacoustic classification: Overcoming training data scarcity using dynamic triplet loss.* J Acoust Soc Am, 2019. **146**(1): p. 534.
40. Szegedy, C., et al., *Going deeper with convolutions*. 2015. 1-9.
41. Nair, V. and G.E. Hinton, *Rectified linear units improve restricted boltzmann machines*, in *Proceedings of the 27th International Conference on International Conference on Machine Learning*. 2010, Omnipress: Haifa, Israel. p. 807–814.
42. Schroff, F., D. Kalenichenko, and J. Philbin. *FaceNet: A unified embedding for face recognition and clustering*. in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.
43. Kingma, D. and J. Ba, *Adam: A Method for Stochastic Optimization*. International Conference on Learning Representations, 2014.
44. Doran, G., *PyEMD: Earth Mover's Distance for Python*. 2014.
45. Ning, Z.-Y., H. Honing, and C. ten Cate, *Zebra finches (Taeniopygia guttata) demonstrate cognitive flexibility in using phonology and sequence of syllables in auditory discrimination*. Animal Cognition, 2023. **26**(4): p. 1161-1175.
46. Virtanen, P., et al., *SciPy 1.0: fundamental algorithms for scientific computing in Python*. Nature Methods, 2020. **17**(3): p. 261-272.