universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Investigating the structure of the greylag goose vocal repertoire: what can unsupervised methods tell us?

verfasst von | submitted by

Lena Gies B.Sc.

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 878
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Verhaltens-, Neuro- und Kognitionsbiologie
Betreut von | Supervisor:	Univ.-Prof. William Tecumseh Sherman Fitch PhD

Acknowledgements

My sincere thanks go to prof. Tecumseh Fitch for his intensive mentoring and teaching, patience and advice, as well as his trust in me. The same extends to prof. Sonia Kleindorfer, whom I also thank for introducing me to the greylag geese. I am very grateful to have such an amazing team of supervisors.

I want to thank Jeroen Van der Aa for giving so many hours of his time to help me put structure into my convoluted thoughts. Jonas Lesigang not only collected a huge portion of the dataset but also infected me with his infinite motivation and curiosity. Thank you to Dr. Tim Sainburg for his suggestion to apply Leiden community detection, his advice on details of the methods and his time. I would like to thank the research teams of the Fitch lab as well as the whole team of the Konrad Lorenz Research Center, notably Dr. Jozsef Arato and Dr. Christian Herbst for helpful insights, advice and ideas. Moreover, talks with prof. Barbara Klump, Lutz Wehrland, and Christian Walter offered support in many ways. Thank you!

Thank you to Alper Yelimlieş for his support in every way and his patience, for many inspiring discussions, shared dreams, and for being the best teammate I could imagine. I would also like to thank my family and friends for their support, for being patient and for being there.

Finally, to take all non-human collaborators into account I would like to thank the greylag geese at the Konrad Lorenz Research Center for their cooperation as well as Fred for his patience when being used as a rubber duck.

Ethical Statement

This study complies with all current Austrian laws and regulations and was supported by Animal Experiment License Number 66.006/0026-WF/V/3b/2014 issued by the Austrian Federal Ministry for Science and Research (EU Standard, equivalent to the Animal Ethics Board). All data collected for this study were obtained using minimally invasive recording and observation methods that did not involve any animal handling. Birds were free-moving and could move away or fly away at any time.

Abstract

Defining a comprehensive signal repertoire is an important step to understanding a vocal communication system. In this thesis, I investigate the vocal repertoire of a well-investigated model system in ethology: the greylag goose (*Anser anser*). I used a large dataset of vocalisations collected over the last four years from a free-living population of greylag geese to investigate the acoustic structure of this species' vocal signals. To this end, I extracted four different types of data representations, which were projected into two dimensions using UMAP, and then clustered using two commonly used methods. In addition, I applied a graph-based clustering approach — Leiden community detection — which, to my knowledge, has not previously been used in bioacoustics. The analyses revealed a partly graded vocal repertoire broadly matching early descriptions of the calls present in the dataset. Audio feature vectors, rather than more commonly used spectrographic representations, revealed clusters most congruent with human labels and offered the most detailed visualisation of the acoustic space. Leiden performed comparably to established approaches but matched the number of human-defined classes closest, with less variable results. These findings highlight the impact that data representation can have in repertoire analysis and provide a quantitative characterisation of the greylag goose vocal repertoire.

Zusammenfassung

Die Definition eines umfassenden Signalrepertoires ist ein wichtiger Schritt zum Verständnis eines vokalen Kommunikationssystems. In dieser Arbeit untersuchte ich das Rufrepertoire eines häufig untersuchten Modellsystems in der Ethologie: der Graugans (*Anser anser*). Ich nutze dazu einen großen Datensatz von Vokalisationen, die in den letzten vier Jahren von einer freilebenden Population von Graugänsen aufgenommen wurden, um die akustische Struktur der Lautsignale dieser Art zu untersuchen. Zu diesem Zweck extrahierte ich vier verschiedene Datendarstellungen, die ich mit Hilfe von UMAP in zwei Dimensionen projizierte und mit zwei gängigen Methoden clusterte. Darüber hinaus wandte ich einen graphenbasierten Clustering-Ansatz - Leiden Community Detection - an, der meines Wissens bisher noch nicht in der Bioakustik verwendet wurde. Meine Analysen ergaben ein teilweise abgestuftes Gesangsrepertoire, das weitgehend mit frühen Beschreibungen der Rufe im Datensatz übereinstimmt. Statt häufiger verwendeten spektrografischen Repräsentationen ergaben Audio-Feature-Vektoren Cluster, die am stärksten mit den menschlichen Bezeichnungen übereinstimmen, und erlaubten die detaillierteste Visualisierung des akustischen Raums. Leiden schnitt vergleichbar mit etablierten Ansätzen ab, entsprach aber der Anzahl der vom Menschen definierten Klassen am ehesten, wobei die Ergebnisse außerdem weniger variabel waren. Diese Ergebnisse verdeutlichen den Einfluss, den die gewählte Darstellung der Daten in der Untersuchung von Lautäußerungsrepertoires haben kann, und bieten eine quantitative Charakterisierung des vokalen Repertoires der Graugans.

1. Introduction

The study of non-human communication systems presents a multitude of challenges, starting with the definition of their signals. While trying to understand these systems, for which first-hand knowledge and insight into the *Umwelt* is limited, researchers have traditionally drawn conclusions based on extensive behavioural observations, which are highly dependent on our own sensory and lived experiences. This subjective element increases the likelihood of observer bias [1], and partial or invalid conclusions. Manually scanning large amounts of data can reduce bias to some extent [2] but can require impractical amounts of time. Modern machine learning (ML) approaches provide a possible solution, potentially allowing less biased and more comprehensive explorations of large datasets for the study of animal communication.

In the last years, ML methods have been increasingly used to study non-human animal communication to determine signal boundaries and classes, with a strong emphasis on the vocal modality. The term ML covers a variety of methods that have been useful for analysing bioacoustic data. For example, Martin and colleagues [3] combined latent projections of a distance matrix of calls with a hierarchical clustering algorithm, which allowed them to analyse a large dataset of rook vocalisations (*Corvus frugilegus*) and uncover different repertoires for each sex. Elie and Theunissen [4] analysed the zebra finch (*Taeniopygia guttata*) vocal repertoire based on four different data representations using supervised and unsupervised algorithms. This approach permitted the authors to quantify the acoustic characteristics of the different call types in detail. Goffinet and colleagues [5] showed that features taken from the latent space of a variational autoencoder (VAE, a deep learning approach to encoding and decoding data) [6] trained on spectrograms outperform hand-picked features when analysing zebra finch and mouse (*Mus musculus*) vocalisations.

But, as with any tool, ML methods only provide meaningful results when applied to the right task in an appropriate manner.

The outcome of ML analyses may depend on the representation of the data, the size of the data set, and the selected parameters and algorithms [7,8]. Sainburg, Thielk, and Gentner [7] reviewed common dimensionality reduction and clustering algorithms, identifying combinations of Principal Component Analysis (PCA) and Uniform Manifold Approximation and Projection (UMAP), which are used to reduce the number of dimensions of the data, with Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), a hierarchical clustering algorithm, to produce clusters best matching hand-labelled call classes in two songbird species. Stowell [9] examined frequently used deep learning architectures and points to the potential of newer Transformer models that include an 'attention' layer as well as automated model design as with the Python framework AutoKeras for classification tasks [10]. However, only a few studies have reviewed the role of input data preprocessing and representation on model performance [11–13].

The vocal communication of the greylag goose (*Anser anser*) is a promising application area for ML methods. The species is highly social, vocally active, and has been a model system in ethology since the field's early days [14]. Greylag geese are waterfowl of the family Anatidae, which have an anatomically simple vocal apparatus with a mostly straight trachea and slight sexual dimorphism [18,29,30]. Greylag males typically vocalise at a higher pitch than females. Würdinger [18] found that females have asymmetrical tympaniform membranes, with larger inner membranes than outer ones, in contrast to males. Additionally, males were found to have a smaller clavicular air sac than females. Different anatomical studies and experiments with excised vocal apparatuses in other *Anser* species determined the

tympaniform membranes to be the source of vocalisations in the genus [18]. Würdinger [18] found the stress on the tympaniform membranes and resulting fundamental frequency to be related to the pressure in the clavicular air sac. Würdinger found multisyllabic calls to have lower fundamental frequencies, ascribing this to the distributed pressure of the clavicular air sac over the longer vocalisation interval. In addition to the fundamental frequency, the formant distribution can be adjusted via the tracheal musculature, which alters the length of the trachea [18,31].

The vocal characteristics of the greylag goose have been described impressionistically since the beginning of the 20th century. Heinroth [15] first described seven different call types of the adult greylag goose in 1910: (1) The distance call ("Lockton") is described as a nasal "gagagag" with an emphasis on the first syllable. This call is described as having individual vocal differences and is produced when partners or families seek out each other from greater distances or without visual contact. (2) The contact call ("Unterhaltungston" or "Stimmfühlungslaut" in [18]) is described as a nasal "gangangang" of 3 to 7 syllables and produced in close proximity to the partner when foraging or walking on foot. (3) Heinroth thought the recruitment call (English name given by [20]) derived from the contact call with rising arousal of the caller: A call similar to the contact call, but with a more distinct syllabic pattern, that is emitted before taking flight. The recruitment call is often described to be accompanied by a distinct head shaking behaviour (e.g. [17]). (4) A call that is produced before a greater distance is covered on foot is described by Heinroth as a quickly repeated "djirb-djarb". Lorenz [16] described this call in terms of context but does not give an auditory description, naming it the locomotion call. (5) The alarm call ("Schreckruf") is described as a short, nasal "gang" that is often followed by the flock fleeing to a nearby body of water. (6) "Hissing", the sixth call that Heinroth described, which is elicited when a goose is very

closely approached, is a broadband nonvocal sound. (7) Finally, the triumph call, a 'blaring' that is repeated continuously and turns into a less intensive nasal "gangangangang", which is produced when an individual returns to their mate after displacing a conspecific.

Fischer [17] calls Heinroths triumph call "rolling" call and describes it as a combination of contact and distance calls, that is not only produced on return to the partner but also when displacing a conspecific. Fischer described an additional call that is uttered after the partners have reunited in triumph: "Cackling", described as fast polysyllabic bout calls of varying loudness. Fischer further lists nest calling as a vocalisation that includes alarm and locomotion as well as rolling calls. She places the adult call types on a spectrum of rising intensity between the contact call and pressed cackling. Fischer distinguishes cackling from pressed cackling through the posture of the neck, which is described to be bent concavely to the ground. Lorenz differentiates the contact call from the greeting call, described as a higher intensity contact call with a distinct posture and adds several other call types from impressionistic observations (e.g. [16]). Overall, Lorenz [14,16] impressionistically described roughly 11 mostly multisyllabic calls in terms of structure and context, with the number of syllables being given considerable weight. Later, Würdinger [18], ten Thoren and Bergmann [19], and others [20] (as cited in [19]) added to these descriptions.

More recently, both experimental playback and more quantitative behavioural investigations into the species' vocal identity have been conducted: Guggenberger and colleagues [21] found that the distance call has individually different call structures and is recognised by the caller's partner in a playback experiment. Weinhäupl [32] found differing call characteristics between individuals' departure calls as well as vocal differences between sexes. Lesigang ([33], in preparation) extends these findings, identifying vocal identity signatures in the

departure call through discriminant function analysis and vocal individual recognition in playback experiments. Körmer [22] identified individual structural variation in the affiliative contact calls. Lesigang and colleagues (in preparation) found that more contact calls were emitted during locomotion and by partnered individuals, with respective partners responding to these calls more often than other members of the flock. Policht and colleagues [34] found vocal identity encoded in hissing vocalisations in the domestic subspecies of *Anser anser*.

Here, I aimed to quantitatively investigate the structure of the greylag goose vocal repertoire using modern ML approaches. I extracted three types of data representations from a manually labelled dataset of syllable-level vocalisation segments with varying loudness: (1) a set of 23 audio feature vectors, (2) linear frequency-scaled spectrograms, and (3) linear frequency cepstral coefficients (LFCC), which quantify the gross shape of the spectrogram. Additionally, I used the computed spectrograms to train a convolutional variational autoencoder (VAE) and used its latent vector as a fourth distinct type of data representation. The dimensionality of all four representation types was reduced to two dimensions using UMAP, before clustering the data with two common approaches: k-means, a non-hierarchical centroid-based clustering algorithm, and HDBSCAN. Additionally, a third, graph-based algorithm was applied to the nearest neighbours graph directly: Leiden community detection [23]. I predicted that automatically detected clusters will match those generated by human observers in terms of the number of classes and the identity of calls within the class. Additionally, I hypothesised that the acoustic space has a graded morphology, in line with descriptions from the early literature [17]. I predicted UMAP projections of spectrograms grouped using HDBSCAN to be most congruent with human labels, as Sainburg and colleagues [7] found in songbirds. The results promise insight into how different approaches to data representation may influence the performance of current ML methods, along with a

comparison of different methods. Finally, to my knowledge, this is the first machine-learning based analysis of a vocal repertoire using Leiden community detection.

2. Materials and Methods

2.1 Study system and site

The study population for this thesis is a free-flying flock of greylag geese (*Anser anser*) at the Konrad Lorenz Research Center in Grünau im Almtal, Austria. The flock was introduced to Grünau im Almtal in 1973 by Konrad Lorenz with long-term research continuously ongoing since then, including individually color-banded and marked birds, and collection of behavioural and life history data, with regularly updated data on individual behavioural differences, social networks, and life histories [24–28].

The flock is non-migratory, habituated to humans, and food-supplemented twice a day at 8 am and between 4 pm and 7 pm at Auingerhof, Grünau im Almtal (47°48’49.7412” N, 13°56’51.72” E). After the morning feeding, individuals fly in groups to the nearby Cumberland Gamepark (47°48’37.6704” N, 13°56’53.9196” E) where they may receive food from visitors of the park. After the afternoon feeding, they move to Lake Alm (47° 45' 12.1356" N, 13° 57' 24.9948" E), Oberganslbach (47° 47' 36.762" N, 13° 56' 57.2316" E), or the Gamepark to sleep. Data collection took place September 2020 to September 2024. Table 1 summarises information about the flock.

2.2 Data collection

The Konrad Lorenz Research Center has gathered an archive of audio recordings containing information on the identity of the calling individual and the attributed call type since 2020. The original dataset collected from this population contained 11,015 WAV files labelled with caller identity, call type, location and time, specifications concerning the recording gear, and sometimes behavioural context. The recordings were collected between September 2020 and

November 2023 at Auingerhof, Oberganslbach and the Cumberland Gamepark. Sampling rates ranged between 44.1 and 48 kHZ at 16-bit-integer to 32-bit-float quantisation.

Because the original dataset had strongly differing sample sizes for the different call types, it was expanded for this study with more targeted audio recordings of alarm and triumph calls. This additional data collection took place in March, August and September 2024 at the locations described above. Vocalisations were recorded in an opportunistic manner at a distance between 2 and 5 meters from the vocalising individual using a Sennheiser MKE 600 directional microphone (Sennheiser electronic SE & Co. KG, Germany) attached to a Zoom F3 (32-bit-float quantisation at 48 kHz) or H5 field recorder (24-bit-integer quantisation at 48 kHz) (Zoom Corporation, Japan).

2.3 Segmentation

All recordings were sub-divided by hand into smaller audio tracks containing a single call type with one or more utterances using Audacity version 3.4.0 (Audacity Team, 2023). The tracks in the database were downsampled to 44.1 kHz to achieve consistent sampling rates, and quantisation was set to 16-bit. Subsequently, the noise of all sound files was reduced through a spectral gating approach using the Python package 'noisereduce' [35] after applying a 0.2 to 16 kHz bandpass filter (SciPy, version 1.14.1, [36]). Since the original segmentation was performed by several researchers with different preprocessing of call segments, new clips were extracted from the noise-reduced original tracks. To make re-segmenting of more than 11,000 calls feasible, this was done semi-automatically, using spectrogram cross-correlation to detect the original segment boundaries in the original track and exporting detected timestamps with the original call type label from the database. Call type annotations as well as segmentations were then manually verified in Raven Lite version 2.0.5 [37]. This approach ensured consistent segment boundaries, preprocessing, as well as

call categorisation. All segments containing overlapping vocalisations, high levels of remaining noise, or unclear call type labels were excluded. Further processing was accomplished using Python (version 3.11.10).

2.4 Preprocessing

Each segment was individually peak normalised to prevent the influence of varying recording distances, although call types may have consistently different loudness levels among them. Then, leading and trailing silences were cut using a threshold of 35 dB SPL down with the *trim* function in the Python package 'librosa' (version 0.10.2, [38]) to ensure consistent segmentation. The final dataset contained single syllables of 6906 calls belonging to six call types in total, with 94 human-labelled as alarm, 1607 contact, 992 departure, 494 distance, 3325 recruitment, and 394 triumph calls of durations between 0.037 and 1.1 seconds. Figure 1 shows example spectrograms of these different call types. This imbalance in number of calls results from some call types being uttered much more frequently, but is also an artefact of data collection targeting specific call types.

2.5 Feature extraction

The data were represented in four ways: (1) vectors of 23 audio features (see below), (2) 64 linear frequency cepstral coefficients (LFCC), (3) Fourier transformed linear frequency spectrograms, and (4) latent vectors of length 128 from a variational autoencoder trained on spectrograms [6]. For the spectrogram-based representation types, the segments were zero-padded logarithmically to the same length of logarithmically scaled durations to account for the large difference in call lengths. For this, the duration of the call segments was log-rescaled and then zero-padded to the length of the longest call segment.

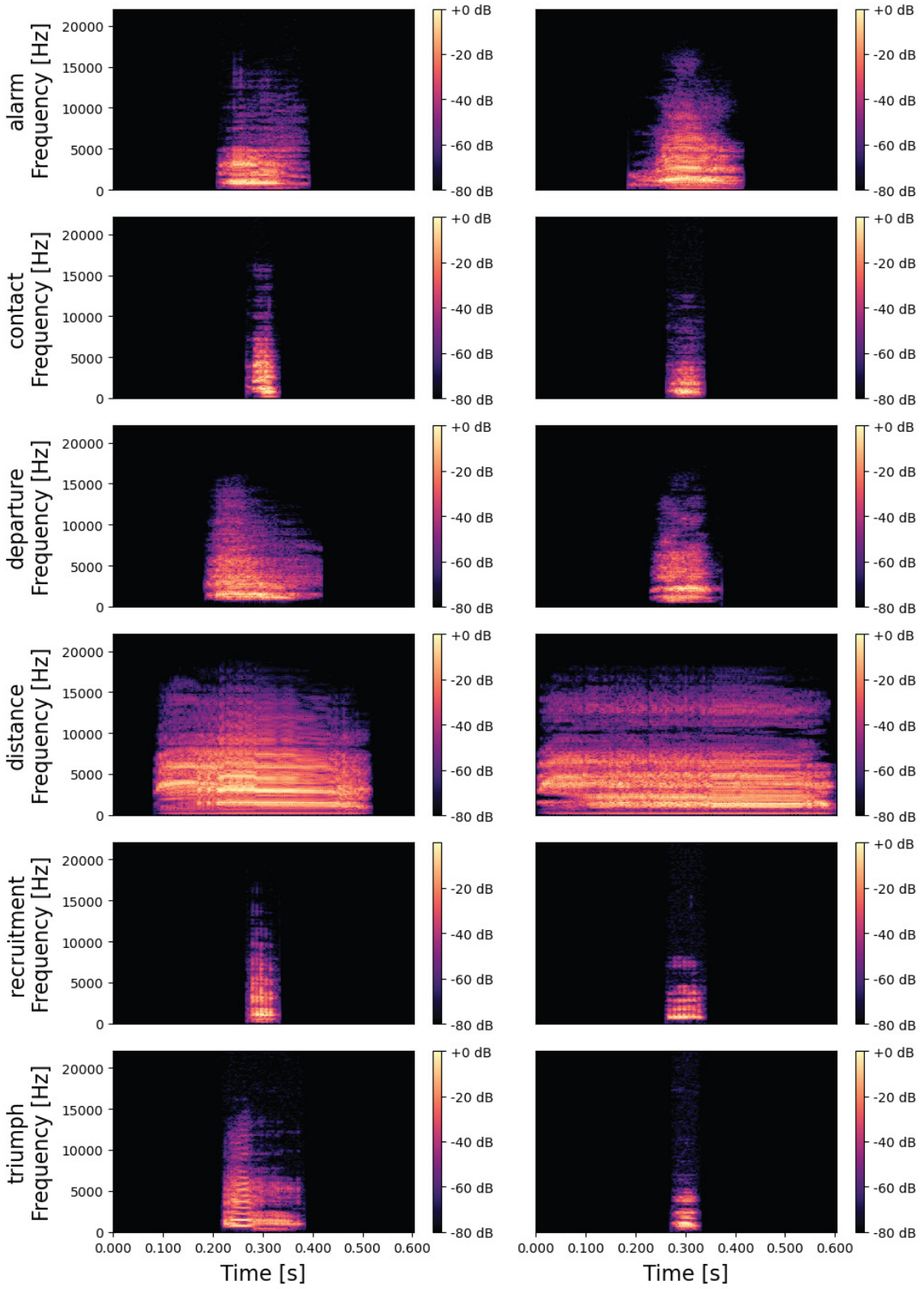

Figure 1 Spectrograms of example calls for the different call type classes. Alarm and departure as well as contact and recruitment calls are structurally very similar. Triumph calls vary in length substantially. Tonality varied internally in some of the classes (e.g. see triumph call examples). Spectrograms were computed using *librosa.stft* with a Hann window of size 512 samples and an overlap of 90%.

2.5.1 Audio Feature Vectors

From the audio segments, eight temporal, nine spectral, and six spectro-temporal features were extracted using the Python packages ‘numpy’ (version 1.26.4, [39]), ‘SciPy’ (version 1.14.1, [36]), ‘librosa’ (version 0.10.2, [38]), and ‘parselmouth’ (version 0.4.5, [40,41]). All extracted features, details on their calculation and the specific python packages used are listed in supplementary table S1.

2.5.2 Linear-frequency Cepstral Coefficients

LFCCs were calculated using the *transforms.lfcc* function of ‘TorchAudio’ (version 2.2.1, [42]) with a maximum frequency of 10 kHz, Hann window of size 512 (12 ms) with a 60% overlap and a linear filter bank of 128 triangular filters. The calculation of the cepstral coefficients quantifies the gross shape of the spectrum, and is well-suited for formant analysis. However, its performance on data with a low signal-to-noise ratio has proven to be disadvantageous [43], and considerable fine spectral structure is discarded through the calculation.

2.5.3 Spectrograms

Linear frequency scale spectrograms were computed using *librosa.stft* with a Hann window of size 512 samples and an overlap of 90%. These spectrograms were transformed to two-dimensional arrays with min-max-scaled pixel values between zero and one. Because little energy was left in the upper 30% of rows, these were cut out, resulting in an array size of 44x170. All of the above data representations were scaled using the class ‘StandardScaler’ of the preprocessing package in the ‘scikit-learn’ Python module (version 1.5.2, [44]) before further analysis.

2.5.4 Variational Autoencoder

Finally, the extracted spectrograms were used to train a convolutional variational autoencoder (VAE) with a latent vector of size 128. Autoencoders [6] are neural networks that learn features, primarily from unlabelled data (unsupervised learning). They encode and decode the original data, mapping it to a lower dimensional latent vector before reconstructing it back to a higher dimension [6]. The reconstruction error, obtained by comparing the input and output layer (usually L2 loss or mean squared error, hereafter MSE), is used to measure the model's performance and adjust the network's weights during training. The latent vector of the trained model can be used as a lower dimensional representation of the data for further analysis. In a VAE, the latent space is modeled using a probabilistic distribution, leading to not one but two latent layers representing mean (μ) and standard deviation (σ) [45]. The latent representation vector z is sampled from this Gaussian distribution. To allow adjusting the network's weights during backpropagation despite this non-differentiable sampling process, the reparameterization trick is used: we do not sample z directly but introduce a random variable ϵ to compute

$$z = \mu + \sigma \cdot \epsilon \tag{1}$$

effectively separating the stochastic sampling from the parameters μ and σ, allowing the adjustment of the gradients during training. The loss function to optimize the model is expanded with the Kullback-Leibler divergence [46], to allow comparing the normal distributions, while the MSE remains as a measure of input and output similarity.

Autoencoders have various applications including noise reduction, generation of new data or, as mentioned above, dimensionality reduction. Several studies in bioacoustics have successfully used autoencoders to reduce the dimensionality of analysed dataset: Best and colleagues [47] find that clusters of datasets for which the dimensionality was reduced with

an autoencoder and UMAP match hand-labelled classes of different species' call classifications closely. Bergler and colleagues [48] classify calls of orcas (*Orcinus orca*) using data reduced with an autoencoder. Rowe et al. [49] use a similar approach, reducing representations extracted from an autoencoder further with t-SNE, to classify bird species from vocalizations. It is worth mentioning that in all the studies above, the autoencoders were trained using spectrograms.

For this study, the VAE was constructed using the package 'PyTorch' (version 2.4, [50]) with a decoder of five convolutional and two fully connected layers (see Supplementary Figure S1 for detailed description). To simplify the layer design, spectrograms were resized to 32x128 pixels using *transforms.resize* of the 'TorchVision' package (version 0.17.1, [51]). To ensure the possibility of future data generation, but weighting a more detailed representation higher than a continuous latent space for clustering, I used the sum of the full MSE plus 10% of the Kullback-Leibler divergence as the loss function.

2.6 Feature embedding

For the clustering pipeline, I employed the entire dataset, along with random subsets of sizes up to 50, 100, 200, and 500 per human-labelled category for every representation type. Due to insufficient sample size, the category 'alarm' was discarded for the subset sizes of 200 and 500. These datasets were randomly drawn and analysed 50 times for each subset size to estimate random error. In addition to the randomly drawn subset sizes, the pipeline was run only five times with the entire dataset, due to computational time constraints.

The dimensionality of the different datasets was reduced for every representation type using UMAP: first, nearest neighbour graphs were extracted based on Euclidean distance using the

Python package ‘umap’ (version 0.5.6, [52]). The argument *n_neighbors* was set to 20% of the subset size or 100 when using the entire dataset. The same package was used to then embed the resulting graphs into two dimensions with the argument *min_dist* = 0, as advised for clustering by [53].

2.6.1 UMAP

UMAP is a feature extraction approach that maps the structure of the original data into a lower dimensional latent space, combining the original features of the data in a nonlinear fashion. Each resulting dimension in the new latent space represents a feature. This allows preserving much of the structure and variance of the original dataset without the need for prior assumptions about the character of the relationships within the data [7]. The algorithm finds a graph approximation of the data, which is then projected into a lower dimensional space using a Riemannian metric by which the data is uniformly distributed in the manifold [52]. To do this, the weights of edges of the graph represent the probability that the nodes are connected. This probability is calculated using a varying radius based on a predefined number of nearest neighbours k, resulting in the parameter k defining how much of the local and global structure will be preserved in the projection of the data. Higher values of k lead to more of the global structure being conserved in the projection. The lower dimensional projection of this graph is optimised using stochastic gradient descent to reduce the cross entropy between the original graph’s structure and that of the projection. The parameter $minDist$ specifies how far apart connected embeddings can be in the embedding space. In comparison to t-SNE [54], another commonly used graph-based dimension reduction algorithm, this holds several advantages: (i) The hyper-parameters k and $minDist$ are more interpretable, (ii) the algorithm preserves more of the global structure, (iii) computation is faster, (iv) reduction of much larger datasets is possible with (v) no restrictions on the original number of dimensions [52] and (vi) the algorithm is less sensitive to noise [7].

2.6.2 Hopkins statistic

The resulting projections of the data were compared to random distributions using the Hopkins statistic as described in Hopkins and Skellam [55] (custom code, Python 3.11.10). The Hopkins statistic compares the distribution of a dataset to that of a randomly sampled dataset and is calculated as

$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i_1}^{m} (u_i^d + w_i^d)} \quad (2)$$

where d is the dimensionality of the data, m is a subset of the data (commonly not higher than 10%, [56]), u_i is the distance of a point in the randomly sampled dataset to its nearest neighbour in the real dataset and w_i is the distance of a point in the real dataset to its nearest neighbour. The randomly sampled dataset spans the space defined by the real dataset. The resulting value lies between 0 and 1. Based on the assumption that distances between nearest neighbours in a clustered dataset will be smaller than those in the randomly sampled dataset, a value close to 1 indicates clustered data whereas a value of 0.5 indicates randomly distributed data. I chose a subset size of 7% of the dataset due to the high number of datapoints.

2.7 Clustering

Subsequently, three clustering algorithms were applied: k-means [57], HDBSCAN [58], and Leiden community detection (hereafter: ‘Leiden’) [23]. Leiden was applied to the nearest neighbours graph directly, whereas k-means and HDBSCAN were used to cluster the two-dimensional UMAP projections.

2.7.1 k-means

The k-means algorithm [57] generates a given number k of clusters by assigning data points to clusters using the mean of each cluster in an iterative manner. k-means was implemented using the Python package 'scikit-learn' (version 1.5.2, [44]). The number of clusters was varied between two and 45, and finally set to obtain the highest silhouette score (see below), while keeping the default values for all other parameters.

2.7.2 HDBSCAN

HDBSCAN [58] is a hierarchical density-based clustering algorithm, based on DBSCAN. The DBSCAN algorithm allows assigning data points to an unidentified number of possibly non-spherical clusters, by calculating the number of neighbouring points in a defined radius *epsilon* and assigning data points above a chosen threshold count of neighbours to a cluster. In HDBSCAN, the radius is varied and the most stable clusters are chosen as the final clusters. I used the Python package 'hdbscan' (version 0.8.39, [60]) to implement the algorithm, setting *min_cluster_size* to 1% of the dataset, leaving the threshold *cluster_selection_epsilon* at 0 to receive the original HDBSCAN results, and optimising the parameter *min_samples* to maintain the highest silhouette score [59] (scikit-learn.metrics, version 1.5.2 [44]).

2.7.3 Leiden community detection

Leiden [23] finds partitions of a graph, optimising its modularity. Modularity compares the number of in-community edges in the graph to the expected value of in-community edges in the same community division with random connections [61]. Leiden was applied using the Python package 'leidenalg' (version 0.10.2, [23]) with partition type *ModularityVertexPartition*. The nearest neighbours graph obtained from UMAP was transformed using 'igraph' (version 0.10.15, [62]).

2.7.4 Silhouette score

I used the silhouette score [59] to assess how well-clustered together data points belonging to the same class were, as well as the most likely number of clusters. To calculate this metric, silhouette coefficients of each data point in the clustered dataset are calculated as follows:

1. Calculate the average euclidean distance between the datapoint and all other data points in the same cluster (a_i)
2. Subtract the euclidean distance of the datapoint to its nearest neighbour in a different cluster (b_i)
3. Divide the resulting value by the maximum value of 1) and 2).

The silhouette coefficient for each data point is calculated as:

$$s_i = \frac{b_i - a_i}{max(a_i - b_i)} \quad (3)$$

The resulting silhouette coefficient ranges between -1 and 1. The value is highest when the distances within the cluster are much smaller than that to the next cluster, implying a good clustering, whereas the data point may lie between clusters, yielding a value around 0. With a value approaching -1 the point would fit the neighbouring cluster better than the one it is currently assigned to. The silhouette score is finally the mean of all silhouette coefficients.

2.7.5 V-measure

All resulting clusters were compared to human labels using V-measure [63] (scikit-learn.metrics, version 1.5.2 [44]), which is the harmonic mean of homogeneity and completeness measures. Homogeneity measures what proportion of clusters contain only data points with the same label, while completeness measures if all members of the same class are also members of the same cluster. Subsequently, the V-measure is then calculated as follows:

$$V = \frac{(1+\beta) \cdot homogeneity \cdot completeness}{\beta \cdot homogeneity + completeness} \quad (4)$$

I chose $\beta = 1$ to weigh homogeneity and completeness equally. V-measure, unlike other metrics like the adjusted Rand index [64], is independent of the number of datapoints, clusters and classes [63].

2.7.6 Adjusted Rand index

The adjusted Rand index (scikit-learn.metrics, version 1.5.2 [44]) was computed in addition to the V-measure, to take another commonly used metric of clustering accuracy into account. As the V-measure, the adjusted Rand index can be used to compare two groupings or human labelled classes to unsupervised classifications. The Rand index can be calculated by taking a pair of data points and comparing their classification in both groupings:

$$R = \frac{a+b}{N} \quad (5)$$

Here, a is the number of data point pairs that belong to the same cluster in both of the compared groupings, b is the number of data point pairs belonging to different clusters in both groupings, and N is the total number of data point pairs [65]. This is adjusted for the possibility of the clusterings overlapping by chance as follows, obtaining the adjusted Rand index:

$$Adj.RS = \frac{RI - \mathbb{E}(R)}{1 - \mathbb{E}(R)} \quad (6)$$

Here, $\mathbb{E}(R)$ is the expected Rand index, if the partitions of the clustering are taken randomly while keeping the same marginal clustering distributions [66].

2.8 Classification

To gain more detailed insight into the acoustic characteristics of the call types as well as to check the validity of the class distinctions, I trained a Linear Discriminant Analysis (LDA)

[67] classifier on the audio feature vectors using 'scikit-learn' with a least-squares solver, leaving all other parameters at their default values. LDA maximises the ratio of between-class to within-class variance by identifying linear combinations of features that separate the given classes. The model was trained on a subset of the original dataset with class sizes capped at 500 of which 20 % were set aside for testing. The f1 score, a common measure for predictive performance, was used for evaluation and tested against chance using a permutation test with 500 permutations and five-fold stratified cross validation with non-overlapping groups. Figure 2 provides a comprehensive overview of the analysis pipeline.

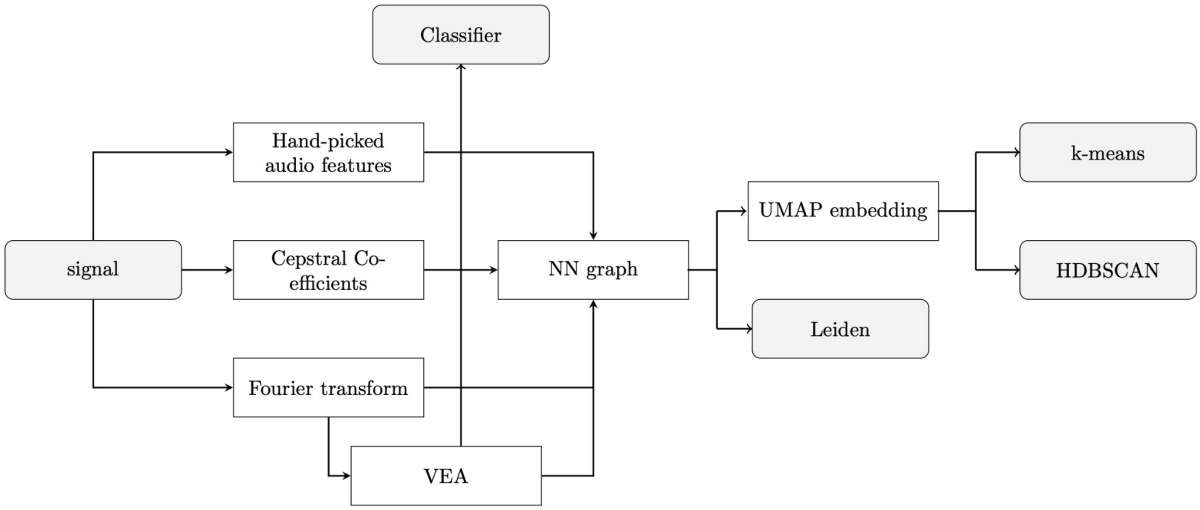

Figure 2: Analysis pipelines for the audio data: Raw data was transformed into linear frequency spectrograms, linear frequency cepstral coefficients and audio feature vectors. Additionally, spectrograms were used to train a variational autoencoder. Of all representations, a nearest neighbours graph was extracted and either used to embed the data into two dimensions using UMAP, or fed into Leiden community detection. All resulting embeddings were then clustered using k-means and HDBSCAN. This pipeline was applied to random subsamples of the original dataset with class sizes of up to 50, 100, 200 and 500 (resampled 50 times each). Graphs were computed based on Euclidean distances and alarm calls were excluded at maximum subset sizes of 200 and 500 due to insufficient sample sizes.

3. Results

3.1 Repertoire structure

The UMAP embeddings of all four data representation types exhibited Hopkins scores indicating non-random distributions (range 0 to 1, with higher likelihood of random distribution at 0). The Hopkins scores for each representation were as follows: audio feature vectors: .94 ± .03, LFCCs: .99 ± .01, spectrograms: .94 ± .02, VAE latent vectors: .97 ± .01 ($H \pm$ SD). While the visualizations appear very different, several regularities are present. Visually, sample embeddings from the contact and recruitment call classes substantially overlapped in all representations except for the audio feature vectors (see figure 3). Similarly, audio segments labelled as triumph calls spread over the entire embedded space, with the exception of the call type 'contact' in embeddings of audio feature vectors, with which generally no overlap was visible. Distance calls were clearly distinctive visually in all embeddings, as was a subgroup consisting of the departure and alarm call classes.

3.2 Call type classification

The six human-assigned call types were classified using LDA. This revealed that the call type classes exhibited the most variation in regards to temporal features (temporal entropy and median, duration, third temporal quartile), as well as spectrographic and power spectral entropy. Figure 4 shows the feature coefficients of the fitted LDA sorted by variance (additionally, see supplementary figures S3 and S4). The model had an overall acceptable performance with an f1 score of .67, which is significantly above chance (p = .002, permutation test, permuted 500 times). Class level f1 scores were as follows: alarm: .17 (support = 19), contact: .64 (support = 100), departure: .82 (support = 100), distance: .91 (support = 99), recruitment: .59 (support = 100), triumph: .50 (support = 79). The LDA was

unable to classify two classes: Only 11 % of alarm calls were assigned correctly, whereas 74 % of them were classified as departure calls. 46 % of triumph calls were correctly labelled. The remaining calls of this category were assigned to the category ‘recruitment’ by 20 %, ‘contact’ by 15 %, ‘departure’ by 11 %, ‘distance’ by 5 %, and ‘alarm’ by 3 %. Moreover, ‘contact’ and ‘recruitment’ classes were interchanged at rates of 63 % against 29 % (human label: contact) and 62 % against 22 % (human label: recruitment). For more details see supplementary figure S2.

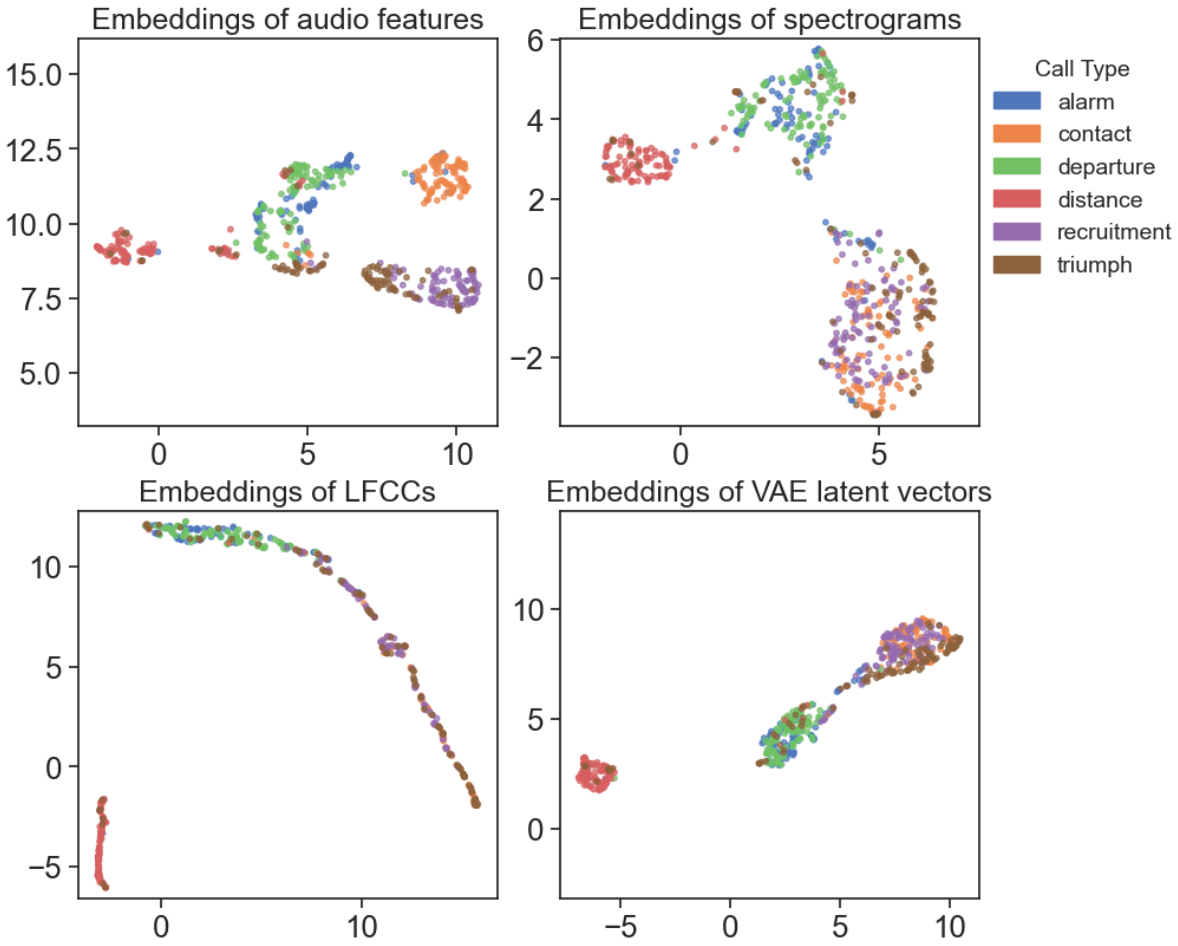

Figure 3 Example UMAP projections of the audio segments based on different representations with subset sizes of up to 100 calls per class. Colours represent hand-labeled class identities. Recruitment (purple) and contact (orange) call categories were only separated in the acoustic space when embedding audio feature vectors (top left). Embeddings of spectrograms and VAE latent vectors (right column) mostly lead to three structural subgroups: (1) mostly distance calls (red), (2) mostly alarm (blue) and departure (green) calls and (3) recruitment and contact calls as well as the biggest proportion of triumph calls (brown). Embeddings of LFCCs (bottom left) mostly lead to distributions approaching one dimension with varying numbers of subgroups.

3.3 Number of predicted call type classes

When clustering the data embeddings with k-means, HDBSCAN and Leiden community detection, the overall mean number of predicted classes was 6.98 ± .15 (mean ± SE, median: 5) overall. The number of clusters detected was highly variable across algorithms for audio feature vectors and LFCC representations, ranging from two to 59 predicted classes, whereas the groupings from spectrogram and VAE vectors ranged between two to seven (spectrogram) and two to ten (VAE vectors) predicted classes (figure 5, supplementary table S2).

Groupings detected with Leiden exhibited less variation (n = 6.53 ± 1.19, mean ± SD) across representations than those detected using HDBSCAN and k-means (HDBSCAN: n = 8.93 ± 11.22, mean ± SD, k-means: n = 5.46 ± 5.73). HDBSCAN classified the data represented by audio feature vectors into more groups than k-means and Leiden, with highly variable results (k-means: n = 4.70 ± 1.21, HDBSCAN: n = 11.53 ± 10.31, Leiden: n = 6.60 ± 0.68, mean ± SD). When clustering LFCCs, both k-means and HDBSCAN produced highly varying results for the different dataset sizes, in contrast to Leiden (k-means: n = 11.87 ± 8.48, HDBSCAN: n = 18.15 ± 15.39, Leiden: n = 7.91 ± 1.14, mean ± SD). All other results for the k-means clusters ranged from two to four (spectrograms) or two to three (VAE latent vectors) groups. Leiden predicted between six and nine communities overall for audio feature vectors as well as five to seven with both spectrograms and VAE latent vectors.

Silhouette scores of the clustered data, which measure how well clustered together data points of the same class are, exceeded zero for all representations, but stayed well below the possible maximum of one: audio feature vectors: .23 ± .02, LFCCs: .37 ± .02, spectrograms: .21 ± .01, VAE latent vectors: .25 ± .03 (s ± SD, range: -1 to 1). Similarly, modularity values for communities detected using Leiden were as follows (range 0 to 1): audio feature vectors:

.68 ± .01, LFCCs: .78 ± .02, spectrograms: .63 ± .01, VAE latent vectors: .66 ± .02 ($Q \pm$ SD).

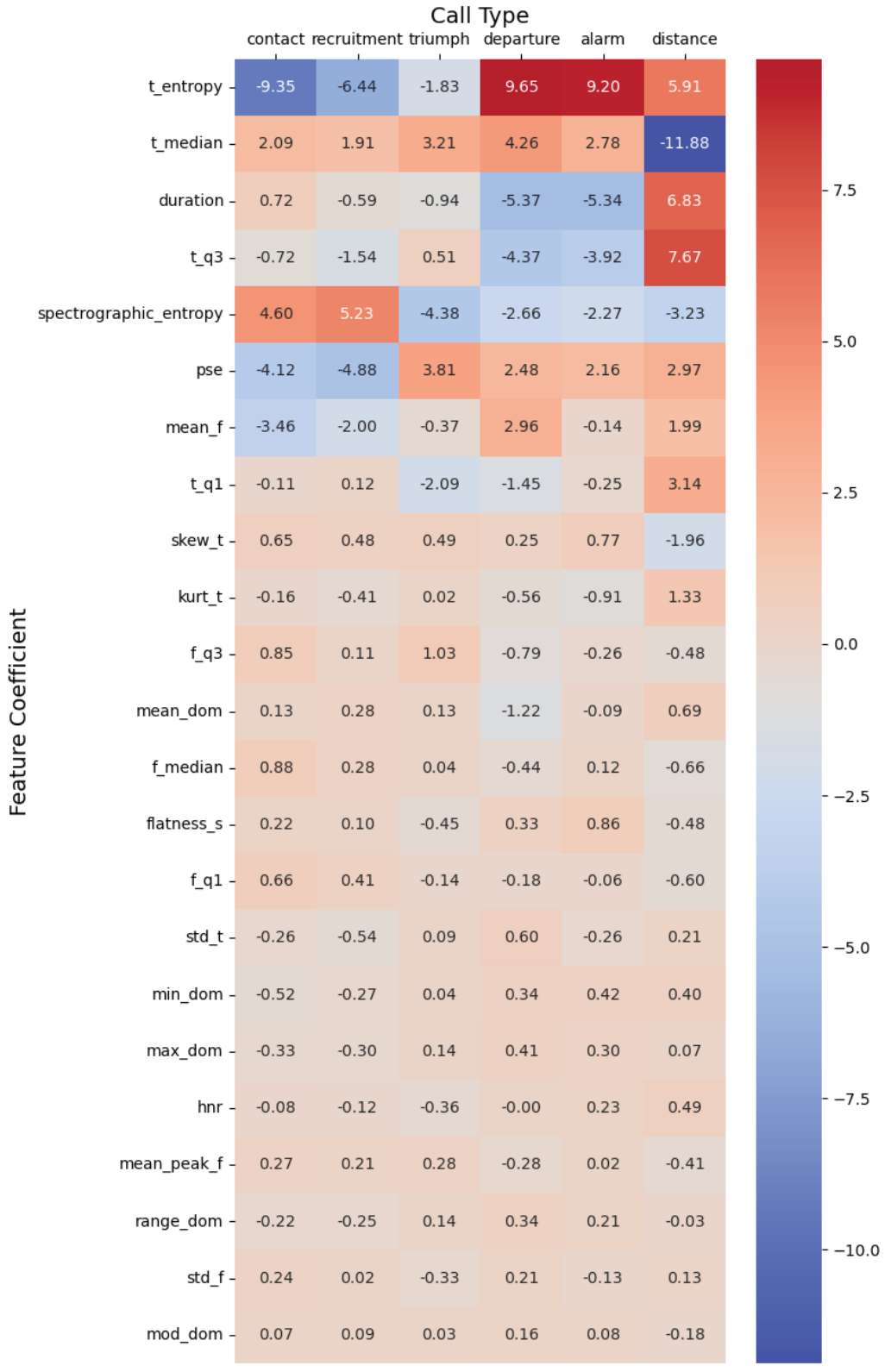

Figure 4 Feature coefficients per call type category obtained from LDA and sorted by variance.

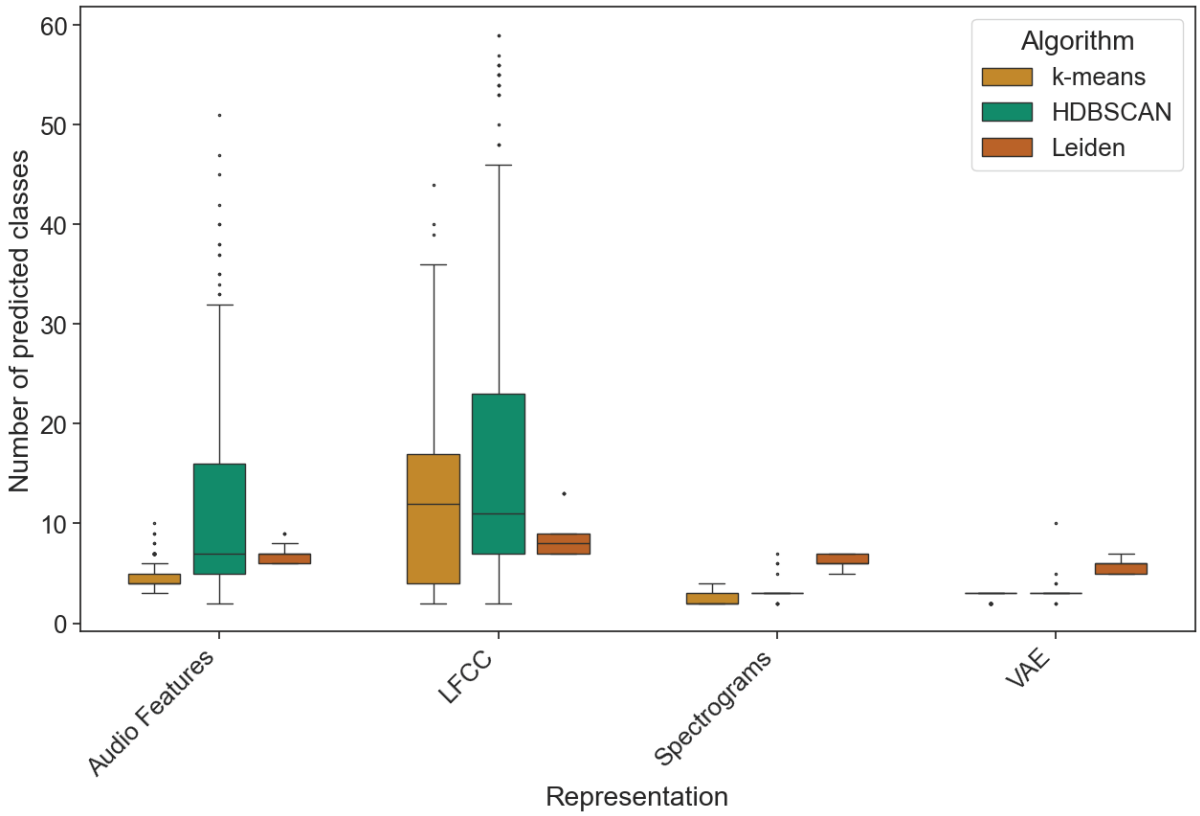

Figure 5 Number of predicted classes for different data representations and algorithms. Whiskers indicate data lying within 1.5 interquartile ranges of the upper and lower quartiles. The data was resampled 255 times in total with five different subset sizes for every representation and algorithm.

3.4 Overlap between predicted and human-labelled classes

The overlap between predicted and hand labelled classes was highest when using audio feature vectors (range 0 to 1, V-measure: .59 ± .08, range -0.5 to 1, adj. RS: .50 ± .12) and VAE vectors (V-measure: .53 ± .05, adj. RS: .40 ± .06). Figure 6 and supplementary table S3 show V-measure and adjusted Rand scores for all representations and algorithms. These metrics varied substantially depending on the subset sizes when using k-means and HDBSCAN to cluster audio feature vectors or LFCCs, whereas the overlap was less variable when detecting groups based on the nearest-neighbours graph directly using Leiden. For spectrogram-based representations, this had the exception that the congruence with known

labels dropped when clustering the entire, unbalanced dataset with Leiden compared to balanced subsets of the original dataset (see figure 7). On the other hand, varying dataset sizes influenced the results less for spectrograms and VAE vectors when using k-means and HDBSCAN. Furthermore, the overlap between hand labelled and predicted classes increased with bigger subset sizes when using k-means or HDBSCAN to cluster audio feature vectors.

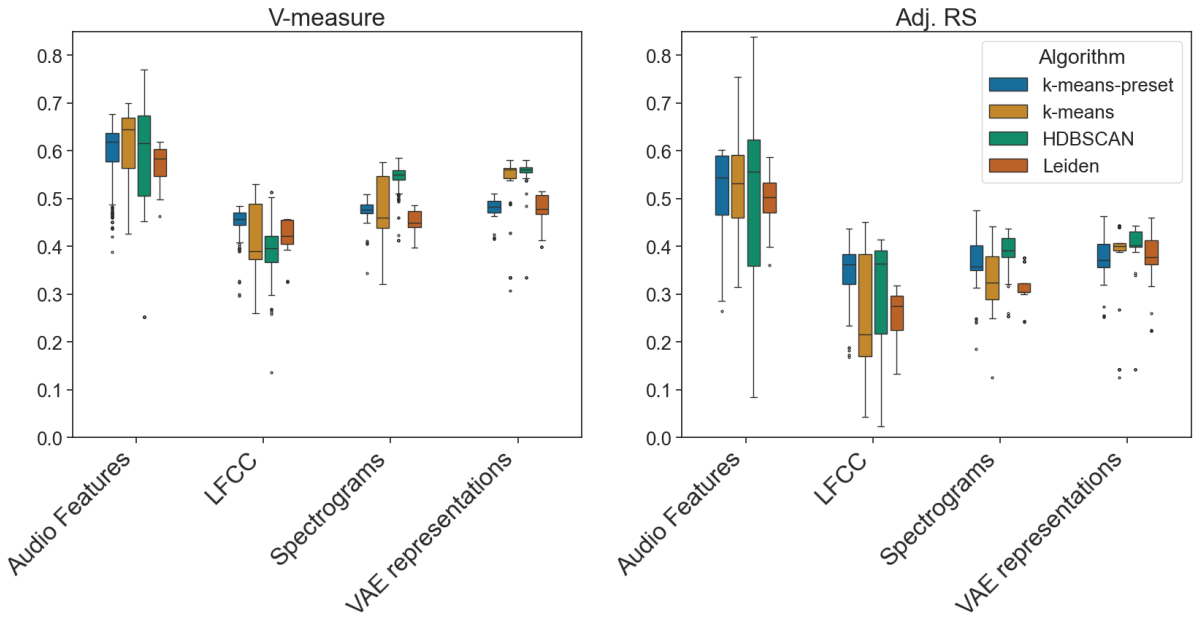

Figure 6 Overlap between hand-labelled and detected class identities for different data representation types and algorithms. Whiskers indicate data lying within 1.5 interquartile ranges of the upper and lower quartiles. The data was resampled 255 times in total with five different subset sizes for every representation and algorithm. V-measure scores (left) can range from zero to one, with one indicating a perfect overlap between human-labelled and predicted classes. Adjusted Rand score (Adj. RS) ranges from -.5 to 1 with perfect alignment of human-labelled and predicted classes at a score of 1.

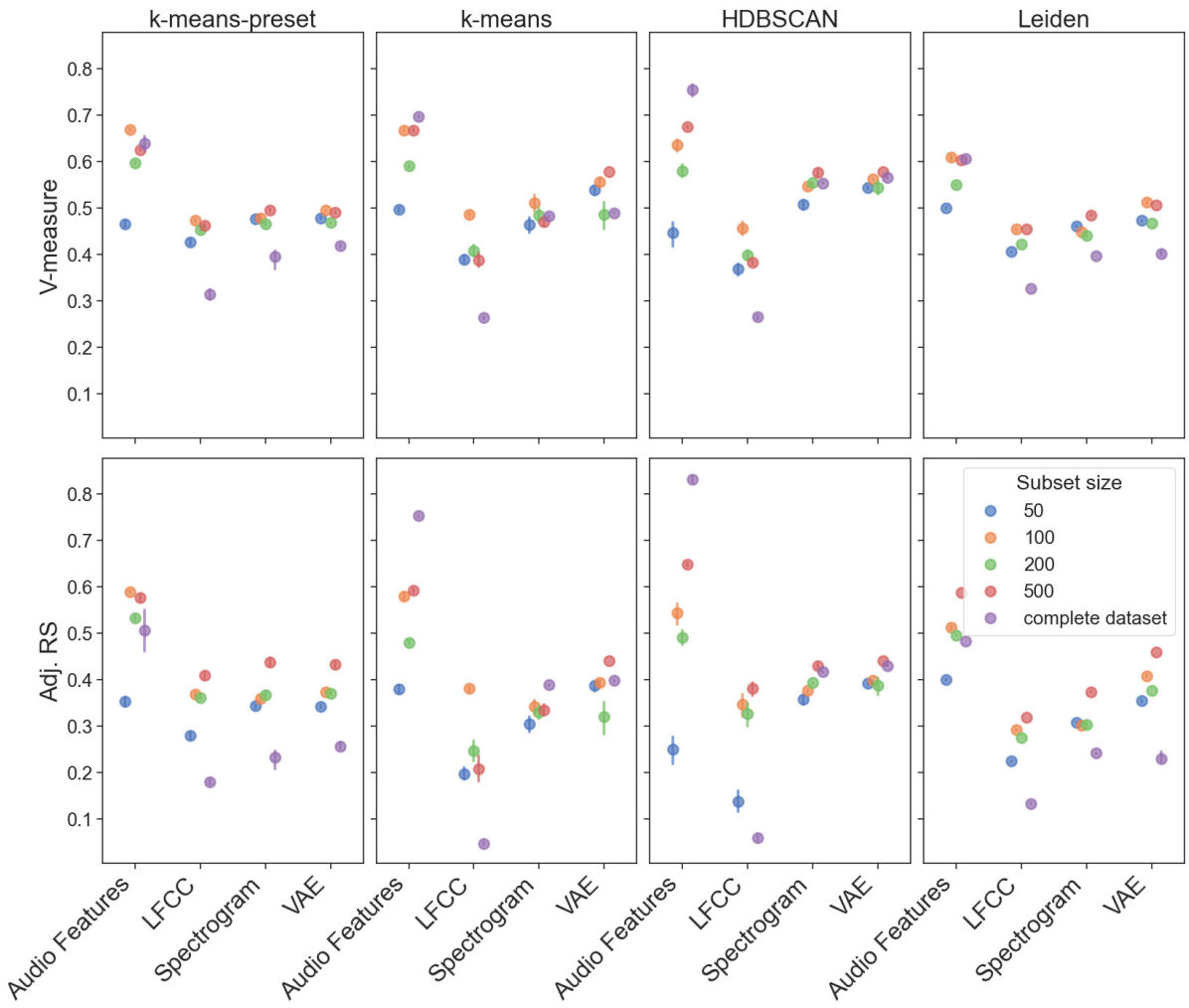

Figure 7 V-measure and adjusted Rand scores with 95 % confidence intervals for different data representation types and clustering algorithms colour coded by maximum subset size. Alarm calls were excluded at maximum subset sizes of 200 and 500. The entire dataset included class sizes ranging from 94 to 3325 and was only run five times, in contrast to 50 times with the randomly resampled subsets of predefined sizes.

4. Discussion

I used unsupervised ML methods to investigate the vocal repertoire of the greylag goose. The visualisation of the acoustic space revealed a partly graded and overlapping vocal repertoire. Distance calls were clearly structurally distinct from other call types. A second distinct group of structurally similar alarm- and departure calls was found. Additionally, contact, recruitment and triumph calls occupied overlapping, but not identical acoustic spaces. Automatically predicted class numbers as well as the overlap between clustered and human-labelled classes varied substantially for the different algorithms and data representations. In the following sections, I first describe the structure of the greylag goose vocal repertoire. I will then outline conclusions regarding the different analysis methods.

4.1 The greylag goose vocal repertoire

4.1.1 Repertoire structure

UMAP embeddings of the data revealed a partly graded vocal repertoire. This is supported by low silhouette and modularity scores as well as low f1 scores for the triumph call category and confusion between recruitment and contact calls when classifying these call types using LDA. Contact and recruitment calls partly overlap in the acoustic space. This is most prominent in spectrogram-based data representations and could be due to most weight being put on the call duration when embedding the visual data, which is moderated when embedding audio feature vectors, but is also apparent when investigating LDA coefficients further: The two call types differ mostly in temporal entropy, duration and mean frequency, but otherwise share very similar feature coefficients. From personal observations, recruitment calls are generally louder than contact calls and have a more isochronous rhythm, which is not taken into account here due to only analysing syllable-level segments and should be

quantitatively investigated. Similarly, departure and alarm calls share very similar acoustic structure, mostly differing in mean frequency and temporal median, and visually overlap in all of the embeddings. This is reflected in a low f1 score when classifying alarm calls, although it is important to note the very limited amount of training data. Triumph calls overlap contact and recruitment calls but span into the departure and alarm call subgroup in terms of their acoustic structure. This supports Fischer's [17] hypothesis that calls emitted in the triumph ceremony fall into two graded classes, which she describes as a combination of contact and distance calls. Distance calls overlap least with all other classes in temporal features and are consistently structurally distinct. This call type is the only one in the dataset that is used with little or no visual contact. It has been hypothesised that graded signals evolved in species with visual contact ([68], as cited in [69]), which may be a possible explanation for this call type's distinctness. Being able to identify the caller from their vocalisation is more important without visual contact.

4.1.2 Number of classes

The embedding of the data consistently resulted in three to four visually largely separate groups, when using representations other than LFCCs, consisting primarily of samples from the following classes: (1) 'contact', (2) 'recruitment' and 'triumph', (3) 'departure' and 'alarm', and (4) 'distance'. Groups one and two were only distinct when embedding audio feature vectors. The structural overlap of triumph as well as alarm calls suggests that these call types can only be discerned when taking other information like the behavioural context, rhythm or sequential context into account. Nonetheless, the human-defined number of call types is in line with the average number of classes suggested by unsupervised methods.

4.2 Methodological insights

To evaluate the performance of different machine learning methodologies, I will compare their output to human call labels, which were made taking behavioral context into account. I found differences in the structure of the embedded space, the number of predicted classes, and the congruence with human labels depending on the chosen data representation type, dataset size, as well as the algorithm chosen when analysing the dataset.

4.2.1 Audio feature vectors

Clusters based on audio feature vectors were most congruent with human labels, independent of the further processing algorithms used. The distinction between recruitment and alarm calls was only evident in embeddings based on audio features. This could be due to audio feature vectors mitigating noise in the data. Furthermore, this observation aligns with findings that audio feature vectors of zebra finch (*Taeniopygia guttata*) vocalisations encode more information than spectrograms [4]. The authors described the analysed vocalisations as "broadband [with a] relatively restricted range of fundamental frequencies", which is true for greylag goose vocalisations as well. However, one study investigating Cassin's vireo (*Vireo cassinii*) song found UMAP embeddings of spectrograms to be more distinctly clustered than those of audio feature vectors [7], contrasting the findings of this thesis. The song of the Cassin's vireo is less broadband than calls of the zebra finch or greylag goose, which may indicate that audio feature vectors are better suited to analyse broadband vocalisations.

4.2.2 Spectrogram-based representations

Among the spectrogram-based data representations, VAE latent vectors had the largest overlap with human labels, specifically when clustered with k-means or HDBSCAN. When clustering spectrograms, groups predicted from HDBSCAN were more congruent with human labels than clusters obtained using all other algorithms, which is in line with findings

by Sainburg and colleagues [7]: they found spectrograms embedded using UMAP and clustered with HDBSCAN to align better with human-labelled data than clusters obtained via k-means in two songbird species. Nevertheless, it may be appropriate to choose VEA latent vectors of spectrograms when analysing vocal data, as previously suggested by Goffinet and colleagues [5]. Although the findings in this thesis differ from the findings of Goffinet and colleagues [5] in that I found audio feature vectors to overlap more with human labels than VAE embeddings overall, VAE embeddings produced better results than spectrograms and LFCCs.

4.2.3 Dataset size

Different balanced input subset sizes, or the use of the unbalanced original dataset, considerably influenced the number of automatically predicted clusters as well as the overlap between clustered and human-labelled groups (see figure 7 and 8). HDBSCAN predicted highly variant numbers of clusters and consequently varying V-measures for audio feature vectors and LFCCs, but not spectrograms or VAE latent vectors, which were influenced the least overall. Groupings from k-means on representations other than LFCCs as well as those based on Leiden with any of the representations were more robust to different dataset sizes in terms of numbers of clusters, but alignment with human labels dropped with the unbalanced original dataset in Leiden groupings based on representations other than audio feature vectors. Congruence with human labels increased with larger dataset sizes when clustering audio feature vector embeddings in both k-means and HDBSCAN, reaching the highest overall V-measure with HDBSCAN applied to the original dataset. To conclude, although results from the combination of Leiden with audio feature vectors varied the least with different dataset sizes, better overlap with human labels could be achieved with the largest dataset size using k-means or HDBSCAN. This was despite the unbalanced class distribution when using the complete dataset.

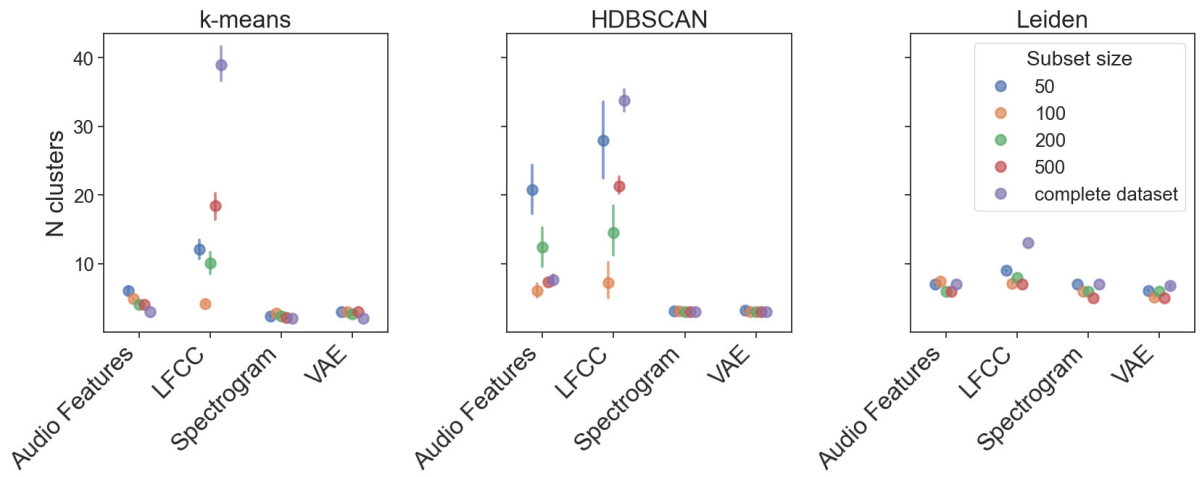

Figure 8 Predicted number of classes with 95 % confidence intervals for different data representation types and clustering algorithms color coded by maximum subset size. Alarm calls were excluded at maximum subset sizes of 200 and 500. The entire dataset included class sizes ranging from 94 to 3325 and was only run five times, in contrast to 50 times with the randomly resampled subsets of predefined sizes.

4.2.4 Clustering algorithms

Although Leiden produced comparable V-measures to the two other algorithms and the number of predicted classes was closest to those originally defined, the overlap with human labels decreased when using Leiden with spectrograms, in comparison to HDBSCAN. The use of audio feature vector embeddings to study vocal repertoires appears to be a sensible choice. The results of this thesis indicate that, for greylag geese, clustering audio feature vectors with HDBSCAN or k-means when the data set is large, or with Leiden if less data is available, produces reasonable results. In some of the projections, Leiden was the only algorithm that was able to distinguish triumph and recruitment calls, indicating it may be well suited for analysing more graded signal repertoires (see figure 9). This study showed that different clustering algorithms produce different results depending on the size of the dataset and the representation of the data. This finding necessitates further closer examination of how these differences correlate with the various parameters, which should additionally include other representation types and distance metrics.

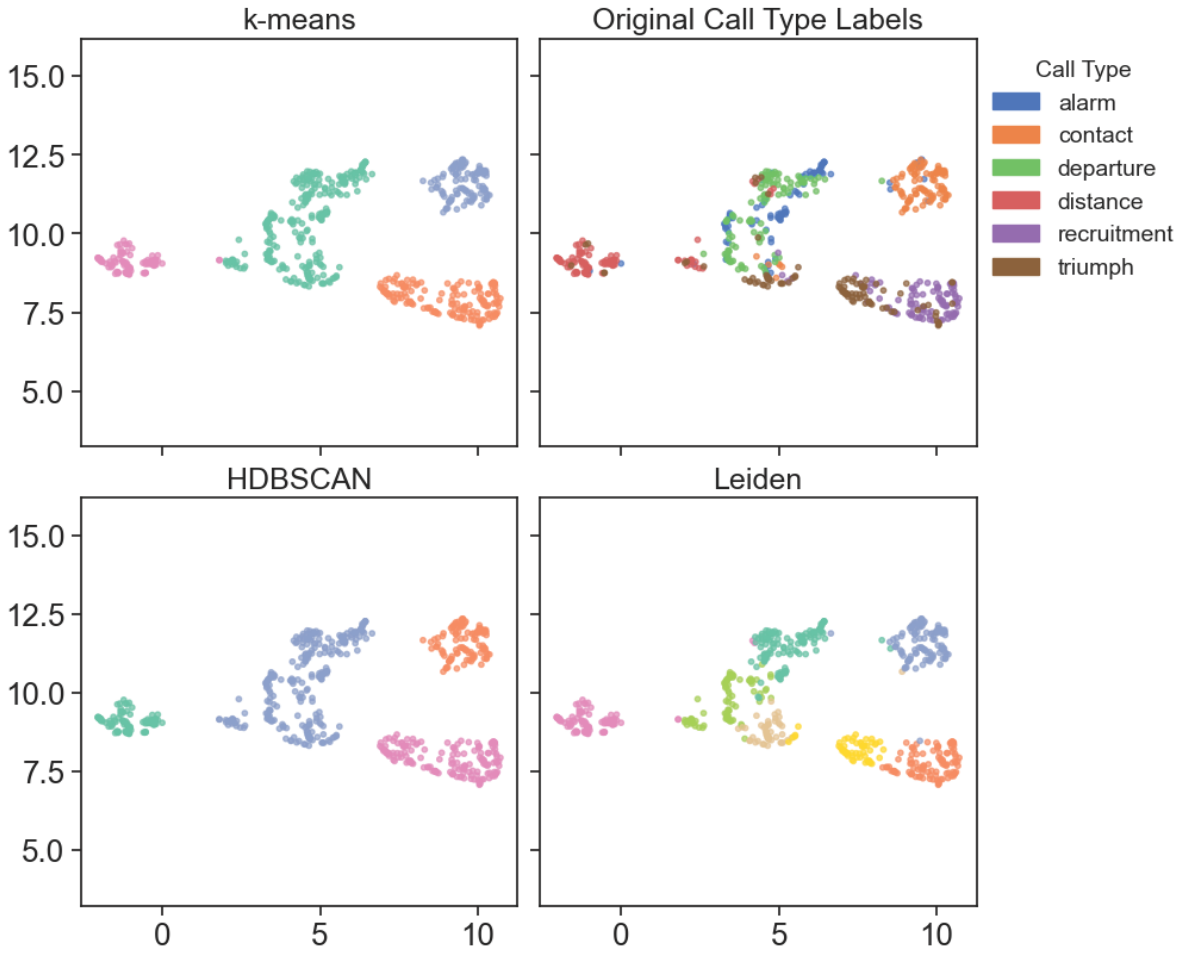

Figure 9 UMAP projections of audio feature vectors. Colours represent human-labelled call types in the upper right plot. Clusters detected using k-means (upper left), HDBSCAN (lower left), and Leiden (lower left) are shown in the remaining plots. Note, that Leiden does not work on the visualised projections directly but the nearest neighbours graph.

4.3 Limitations

This study provides insight into the vocal repertoire of the adult greylag goose, but does not include all of the observed call types. Hisses were excluded due to insufficient data, and call types mentioned in the early literature like the greeting or locomotion call [16] were not included in the data collection process. Lorenz describes both of these call types as higher intensity contact calls, otherwise only distinguishing them by context and posture. Due to personal observations, I suspect these calls to be contextual variants of the contact call. Thus,

this analysis should not be mistaken for an investigation into the size of the greylag goose vocal repertoire, but rather an exploration of its structure.

Additionally, the analysed data were collected in a field setting and still contain some noise even after applying a noise reduction algorithm. Furthermore, recordings were taken at different distances to the vocalising individuals and therefore amplitude normalised. This discards information on relative loudness of call types, but may also leave structural artefacts in the energy distribution of the call, specifically less energy remaining in the higher frequency range for more distant calls, which may lead to distances in UMAP projections that do not reflect the structural differences of the emitted sounds.

Furthermore, I did not investigate the rhythmic and sequential domain of the vocalisations, analysing only single syllables. As the call types have been consistently described in terms including their rhythm and syllable count, this poses a considerable limitation. However, as I already find the individual syllables to match human-labelled classes to a considerable degree, further investigating sequencing and/or rhythm may enable us to distinguish acoustically overlapping call types like the triumph or alarm call.

Methodologically, I employed an algorithm which has, to my knowledge, not yet been used in bioacoustics analyses, Leiden community detection, and found it to produce results comparable to those of more commonly used clustering algorithms. It is important to note that HDBSCAN and k-means clusterings based on two-dimensional UMAP embeddings were compared to Leiden clusterings on the nearest-neighbours graph directly, which may provide Leiden with an advantage, since the additional step of dimension reduction would already exclude some information relevant to cluster detection. In addition, Leiden was found to

generate too many clusters [70]. This should be considered in future work, even though the average number of predicted clusters matched the number of human labels best when using this algorithm. I applied only one partitioning approach for the Leiden clusterings and the graph was not preprocessed before detecting the communities, which provides an opportunity for future improvements.

Nonetheless, these results show that the choice of the data representation types and the algorithm used for investigating a species' vocal repertoire can influence the results considerably. Audio feature vectors allowed a more detailed visualization of the acoustic space than other, spectrogram-based data representation types that were analysed here and lead to clusterings most congruent with human labels. VAE latent vectors obtained from spectrograms produced better overlap of automatically and human predicted classes than other spectrogram-based representation types. All of the three tested clustering algorithms produced good results overall, but there was a high variation in the number of predicted classes, depending on the choice of the data representation type as well as the dataset size and balance when using HDBSCAN and k-means. Leiden community detection matched the human-defined number of classes closest, but the overlap of human-labelled and predicted class identities dropped when using the original, unbalanced dataset. In contrast to this, HDBSCAN and k-means were more robust to the unbalanced dataset and even produced better results with the larger original dataset when using audio feature vectors, despite its unbalanced class distribution. Overall, these results show that unsupervised machine learning methods should be applied with caution in bioacoustics analyses, considering the limitations of the different algorithms and data representation types.

5. Conclusion

The greylag goose has been a model system in ethology since the field's early days, but the species' entire vocal repertoire has not previously been quantitatively investigated. I used different data representations and clustering algorithms to investigate the species' vocal signals, building on years of behavioural observation of individually marked birds, and analysing audio data that have been systematically collected since 2020. I find both graded and distinct signals that are largely congruent with human-labelled signal classes. However, the representation format of the data substantially influenced both the visualisations of the acoustic space and the outcomes of the algorithms used. When investigating vocal repertoires, it is useful to compare different types of data representation, and carefully consider the algorithms used, especially when estimating the number of signal classes. Additionally, I found that the size and balance of the input dataset can have a large influence on the projection of the data. Nonetheless, these analyses illustrate the power of modern machine learning techniques to illuminate the structure of a vocal repertoire, even for well-studied species, and show the promise of applying these analyses to species with less-studied vocal repertoires.

6. References

1. Sackett DL. 1979 Bias in analytic research. *J. Chronic Dis.* **32**, 51–63. (doi:doi.org/10.1016/0021-9681(79)90012-2)

2. Provost KL, Yang J, Carstens BC. 2022 The impacts of fine-tuning, phylogenetic distance, and sample size on big-data bioacoustics. *PLOS ONE* **17**, e0278522. (doi:10.1371/journal.pone.0278522)

3. Martin K, Cornero FM, Clayton NS, Adam O, Obin N, Dufour V. 2024 Vocal complexity in a socially complex corvid: gradation, diversity and lack of common call repertoire in male rooks. *R. Soc. Open Sci.* **11**, 231713. (doi:10.1098/rsos.231713)

4. Elie JE, Theunissen FE. 2016 The vocal repertoire of the domesticated zebra finch: a data-driven approach to decipher the information-bearing acoustic features of communication signals. *Anim. Cogn.* **19**, 285–315. (doi:10.1007/s10071-015-0933-6)

5. Goffinet J, Brudner S, Mooney R, Pearson J. 2021 Low-dimensional learned feature spaces quantify individual and group differences in vocal repertoires. *eLife* **10**, e67855. (doi:10.7554/eLife.67855)

6. Berahmand K, Daneshfar F, Salehi ES, Li Y, Xu Y. 2024 Autoencoders and their applications in machine learning: a survey. *Artif. Intell. Rev.* **57**, 28. (doi:10.1007/s10462-023-10662-6)

7. Sainburg T, Thielk M, Gentner TQ. 2020 Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLOS Comput. Biol.* **16**, 1–48. (doi:10.1371/journal.pcbi.1008228)

8. Wierucka K *et al.* 2024 Same data, different results? Evaluating machine learning approaches for individual identification in animal vocalisations. (doi:10.1101/2024.04.14.589403)

9. Stowell D. 2022 Computational bioacoustics with deep learning: a review and roadmap. *PeerJ* **10**, e13152. (doi:10.7717/peerj.13152)

10. Jin H, Chollet F, Song Q, Hu X. 2023 AutoKeras: An AutoML Library for Deep Learning. *J. Mach. Learn. Res.* **24**, 1–6. (doi:10.5555/3648699.3648705)

11. Napier T, Ahn E, Allen-Ankins S, Schwarzkopf L, Lee I. 2024 Advancements in preprocessing, detection and classification techniques for ecoacoustic data: A comprehensive review for large-scale Passive Acoustic Monitoring. *Expert Syst. Appl.* **252**, 124220. (doi:10.1016/j.eswa.2024.124220)

12. Kvsn RR, Montgomery J, Garg S, Charleston M. 2020 Bioacoustics Data Analysis – A Taxonomy, Survey and Open Challenges. *IEEE Access* **8**, 57684–57708. (doi:10.1109/ACCESS.2020.2978547)

13. Tolkova I. 2021 Feature Representations for Conservation Bioacoustics: Review and Discussion. In *IJCAI 2021 Workshop on AI for Social Good*,

14. Lorenz K. 1935 Der Kumpan in der Umwelt des Vogels. *J. Für Ornithol.* **83**, 137–213. (doi:10.1007/BF01905355)

15. Heinroth O. 1910 Beiträge zur Biologie, namentlich Ethologie und Psychologie der Anatiden.

16. Lorenz K. 1988 *Hier bin ich – wo bist du? Ethologie der Graugans*. Munich: Piper.

17. Fischer H. 1964 Das Triumphgeschrei der Graugans (*Anser anser*). *Z. Für Tierpsychol.* **22**, 247–304. (doi:10.1111/j.1439-0310.1965.tb01498.x)

18. Würdinger I. 1970 Erzeugung, Ontogenie und Funktion der Lautäußerungen bei vier Gänsearten: (*Anser indicus, A. caerulescens, A. albifrons* und *Branta canadensis*). *Z. Für Tierpsychol.* **27**, 257–302. (doi:10.1111/j.1439-0310.1970.tb01875.x)

19. ten Thoren A, Bergmann H-H. 1987 Die Entwicklung der Lautäußerungen bei der Graugans (*Anser anser*). *J. Orn.* , 181–207.

20. Kear J. 1968 The calls of very young Anatidae. *Beih Vogelwelt* **1**, 93–113.

21. Guggenberger M, Adreani NM, Foerster K, Kleindorfer S. 2022 Vocal recognition of distance calls in a group-living basal bird: the greylag goose, *Anser anser*. *Anim. Behav.* **186**, 107–119. (doi:10.1016/j.anbehav.2022.01.004)

22. Körmer E-M. [masters thesis] 2022 The call structure of contact calls in greylag geese (*Anser anser*).

23. Traag VA, Waltman L, Van Eck NJ. 2019 From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 5233. (doi:10.1038/s41598-019-41695-z)

24. Kleindorfer S. 2024 *Die erstaunliche Welt der Graugänse: Wie sie leben, kommunizieren und füreinander sorgen*. Christian Brandstätter Verlag.

25. Kleindorfer S, Krupka MA, Katsis AC, Frigerio D, Common LK. 2024 Aggressiveness predicts dominance rank in greylag geese: mirror tests and agonistic interactions. *R. Soc. Open Sci.* **11**, 231686. (doi:doi.org/10.1098/rsos.231686)

26. Kleindorfer S, Heger B, Tohl D, Frigerio D, Hemetsberger J, Fusani L, Fitch WT, Colombelli-Négrel D. 2024 Cues to individuality in greylag goose faces: algorithmic discrimination and behavioral field tests. *J. Ornithol.* **165**, 27–37. (doi:10.1007/s10336-023-02113-4)

27. Katsis A, Common L, Lesigang J, Bold A, Fröhlich M, Schmincke J-M, Frigerio D, Kleindorfer S. 2024 Flight initiation distance is repeatable and geographically flexible in greylag geese *Anser anser*. *J. Avian Biol.* (doi:10.1111/jav.03288)

28. Common LK, Katsis AC, Frigerio D, Kleindorfer S. 2024 Effects of assortative mating for personality on reproductive success in greylag geese, *Anser anser*. *Anim. Behav.* **216**, 141–153. (doi:https://doi.org/10.1016/j.anbehav.2024.08.004)

29. Johnsgard PA. 1961 Tracheal anatomy of the anatidae and its taxonomic significance. *Wildfowl Trust* **12th Annual Report**, 58–69.

30. Johnsgard PA. 1971 Observations on sound production in the Anatidae.

31. Fitch WT, Anikin A, Pisanski K, Valente D, Reby D. In press. Formant analysis of vertebrate vocalizations: Achievements, pitfalls & promises. *BMC Biol.*

32. Weinhäupl V. [masters thesis] 2022 Departure Calls in Greylag Geese (*Anser anser*). University of Natural Resources and Life Sciences, Vienna.

33. Lesigang J. [masters thesis] 2024 Identity encoding and recognition in greylag goose departure calls. University of Vienna, Vienna.

34. Policht R, Kowalczyk A, Łukaszewicz E, Hart V. 2020 Hissing of geese: caller identity encoded in a non-vocal acoustic signal. *PeerJ* **8**, e10197. (doi:10.7717/peerj.10197)

35. Sainburg T. 2019 timsainb/noisereduce: v1.0. (doi:10.5281/zenodo.3243139)

36. Virtanen P *et al.* 2020 SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272. (doi:10.1038/s41592-019-0686-2)

37. K. Lisa Yang Center for Conservation Bioacoustics at the Cornell Lab of Ornithology. 2024 Raven Lite.

38. McFee B, Raffel C, Liang D, Ellis D, McVicar M, Battenberg E, Nieto O. 2015 librosa: Audio and music signal analysis in Python. pp. 18–24. Austin, Texas. (doi:10.25080/Majora-7b98e3ed-003)

39. Harris CR *et al.* 2020 Array programming with NumPy. *Nature* **585**, 357–362. (doi:10.1038/s41586-020-2649-2)

40. Boersma P, Weenink D. 2021 Praat: doing phonetics by computer.

41. Jadoul Y, Thompson B, Boer B de. 2018 Introducing Parselmouth: A Python interface to Praat. *J. Phon.* **71**, 1–15. (doi:https://doi.org/10.1016/j.wocn.2018.07.001)

42. Hwang J *et al.* 2023 TorchAudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for PyTorch.

43. Abdul ZKh, Al-Talabani AK. 2022 Mel frequency cepstral coefficient and its applications: A Review. *IEEE Access* **10**, 122136–122158. (doi:10.1109/ACCESS.2022.3223444)

44. Pedregosa F *et al.* 2011 Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**.

45. Kingma DP, Welling M. [preprint] 2013 Auto-Encoding Variational Bayes. (doi:10.48550/arXiv.1312.6114)

46. Kullback S, Leibler RA. 1951 On information and sufficiency. *Ann. Math. Stat.* **22**, 79–86.

47. Best P, Paris S, Glotin H, Marxer R. 2023 Deep audio embeddings for vocalisation clustering. *PLOS ONE* **18**, e0283396. (doi:10.1371/journal.pone.0283396)

48. Bergler C, Schmitt M, Cheng RX, Maier A, Barth V, Nöth E. 2019 Deep Learning for Orca Call Type Identification — A fully unsupervised approach. In *Interspeech 2019*, pp. 3357–3361. ISCA. (doi:10.21437/Interspeech.2019-1857)

49. Rowe B, Eichinski P, Zhang J, Roe P. 2021 Acoustic auto-encoders for biodiversity assessment. *Ecol. Inform.* **62**, 101237. (doi:10.1016/j.ecoinf.2021.101237)

50. Ansel J *et al.* 2024 PyTorch 2: Faster machine learning through dynamic Python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pp. 929–947. La Jolla CA USA: ACM. (doi:10.1145/3620665.3640366)

51. maintainers T, contributors. 2016 TorchVision: PyTorch's Computer Vision library. *GitHub Repos.*

52. McInnes L, Healy J, Melville J. 2020 UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. (doi:https://doi.org/10.48550/arXiv.1802.03426)

53. Thomas M, Jensen FH, Averly B, Demartsev V, Manser MB, Sainburg T, Roch MA, Strandburg‐Peshkin A. 2022 A practical guide for generating unsupervised, spectrogram‐based latent space representations of animal vocalizations. *J. Anim. Ecol.* **91**, 1567–1581. (doi:10.1111/1365-2656.13754)

54. van der Maaten L, Hinton G. 2008 Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605.

55. Hopkins B, Skellam JG. 1954 A new method for determining the type of distribution of plant individuals. *Ann. Bot.* **18**, 213–227. (doi:10.1093/oxfordjournals.aob.a083391)

56. Wright K. 2022 Will the real Hopkins Statistic please stand up? *R J.* **14**, 282–292. (doi:10.32614/RJ-2022-055)

57. Krishna K, Narasimha Murty M. 1999 Genetic K-means algorithm. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **29**, 433–439. (doi:10.1109/3477.764879)

58. Campello RJGB, Moulavi D, Sander J. 2013 Density-Based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining* (eds J Pei, VS Tseng, L Cao, H Motoda, G Xu), pp. 160–172. Berlin, Heidelberg: Springer Berlin Heidelberg. (doi:10.1007/978-3-642-37456-2_14)

59. Rousseeuw PJ. 1987 Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65. (doi:10.1016/0377-0427(87)90125-7)

60. McInnes L, Healy J, Astels S. 2017 hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* **2**, 205. (doi:10.21105/joss.00205)

61. Newman MEJ, Girvan M. 2004 Finding and evaluating community structure in networks. *Phys. Rev. E* **69**, 026113. (doi:10.1103/PhysRevE.69.026113)

62. Csárdi G, Nepusz T, Horvát S, Traag V, Zanini F, Noom D. 2024 igraph. (doi:10.5281/zenodo.14044797)

63. Rosenberg A, Hirschberg J. 2007 V-Measure: A conditional entropy-Based external cluster evaluation measure.

64. Rand WM. 1971 Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**, 846–850. (doi:10.1080/01621459.1971.10482356)

65. Chacón JE, Rastrojo AI. 2023 Minimum adjusted Rand index for two clusterings of a given size. *Adv. Data Anal. Classif.* **17**, 125–133. (doi:10.1007/s11634-022-00491-w)

66. Hubert L, Arabie P. 1985 Comparing partitions. *J. Classif.* **2**, 193–218. (doi:10.1007/BF01908075)

67. Balakrishnama S, Ganapathiraju A. 1998 Linear Discriminant Analysis—A brief tutorial. **11**.

68. Marler P. 1975 On the origin of speech from animal sounds In: Kavanagh, Cutting, editors. The Role of Speech in Language.

69. Wadewitz P, Hammerschmidt K, Battaglia D, Witt A, Wolf F, Fischer J. 2015 Characterizing vocal repertoires—Hard vs. soft classification approaches. *PLOS ONE* **10**, e0125785. (doi:10.1371/journal.pone.0125785)

70. Grabski IN, Street K, Irizarry RA. 2023 Significance analysis for clustering with single-cell RNA-sequencing data. *Nat. Methods* **20**, 1196–1202. (doi:10.1038/s41592-023-01933-9)

71. Sueur J. 2018 *Sound analysis and synthesis with R*. Cham: Springer.

8. Supplementary material

Table S1 Extracted audio features and the corresponding python packages used for calculation. All calculations from [71] unless marked with the corresponding function. The dominant frequencies were detected using the function *find_peaks* of the package 'SciPy' using a prominence and height of 5 % of the amplitude range and a minimal horizontal distance of 5 % of the signal length.

Domain	Name	Packages Used	Equation
Temporal	Duration [s]	`scipy.signal`	
	Temporal quartiles [s]	`numpy`	Time values at the 25th, 50th, and 75th percentiles of the cumulative normalized amplitude envelope
	Temporal std. deviation [s]	`numpy`	Standard deviation of the amplitude envelope
	Temporal skew	`scipy.stats`	`skew()`
	Temporal kurtosis	`scipy.stats`	`kurtosis()`
	Temporal entropy	`numpy`	Shannon-Wiener Entropy of the normalized temporal amplitude distribution
Spectral	Mean frequency [kHz]	`numpy`	
	Mean peak frequency [kHz]	`numpy`	Frequency corresponding to the maximum value in the power spectrum
	Spectral std. deviation [kHz]	`numpy`	Standard deviation of frequencies weighted by power spectrum
	Spectral flatness	`scipy.stats, numpy`	Geometric mean over arithmetic mean of the power spectrum
	Spectral quartiles [kHz]	`numpy`	Frequencies at the 25th, 50th, and 75th percentiles of cumulative spectral energy
	Harmonics-to-noise ratio [dB]	`parselmouth`	`Sound.to_harmonicity()`
	Power spectral entropy	`numpy`	Shannon-Wiener Entropy of the normalized power spectral distribution
Spectro-temporal	Spectrographic entropy	`numpy`	Product of temporal and spectral entropy
	Mean dominant frequency [kHz]	`librosa, scipy.signal, numpy`	
	Min dominant frequency [kHz]	`librosa, scipy.signal, numpy`	
	Max dominant frequency [kHz]	`librosa, scipy.signal, numpy`	
	Dominant frequency range [kHz]	`librosa, scipy.signal, numpy`	
	Dominant frequency modulation [kHz]	`librosa, scipy.signal, numpy`	Cumulative difference of the dominant frequency over its range

Table S2 Average number and range of predicted groups per clustering algorithm.

Algorithm	**Number of clusters (range)**	**Mean ± SD**
k-means	2 - 44	5.463 ± 5.727
HDBSCAN	2 - 59	8.934 ± 11.221
Leiden	5 - 13	6.529 ± 1.190

Table S3 Overlap between hand-labelled and automatically clustered classes.

Representation	**mean V-measure ± SD**	**mean adj. RS ± SD**	**Algorithm**	**V-measure ± SD**	**Adj. RS ± SD**
PAF	.588 ± .080	.502 ± .119	k-means	.607 ± .072	.514 ± .096
			HDBSCAN	.589 ± .105	.492 ± .170
			Leiden	.567 ± .045	.499 ± .066
LFCC	.415 ± .051	.274 ± .099	k-means	.414 ± .059	.254 ± .105
			HDBSCAN	.398 ± .055	.293 ± .128
			Leiden	.432 ± .027	.275 ± .041
Spectrograms	.495 ± .054	.347 ± .052	k-means	.483 ± .056	.330 ± .054
			HDBSCAN	.547 ± .032	.391 ± .035
			Leiden	.457 ± .020	.320 ± .033
VEA embeddings	.528 ± .053	.396 ± .057	k-means	.539 ± .065	.386 ± .077
			HDBSCAN	.557 ± .030	.406 ± .038
			Leiden	.488 ± .024	.396 ± .047

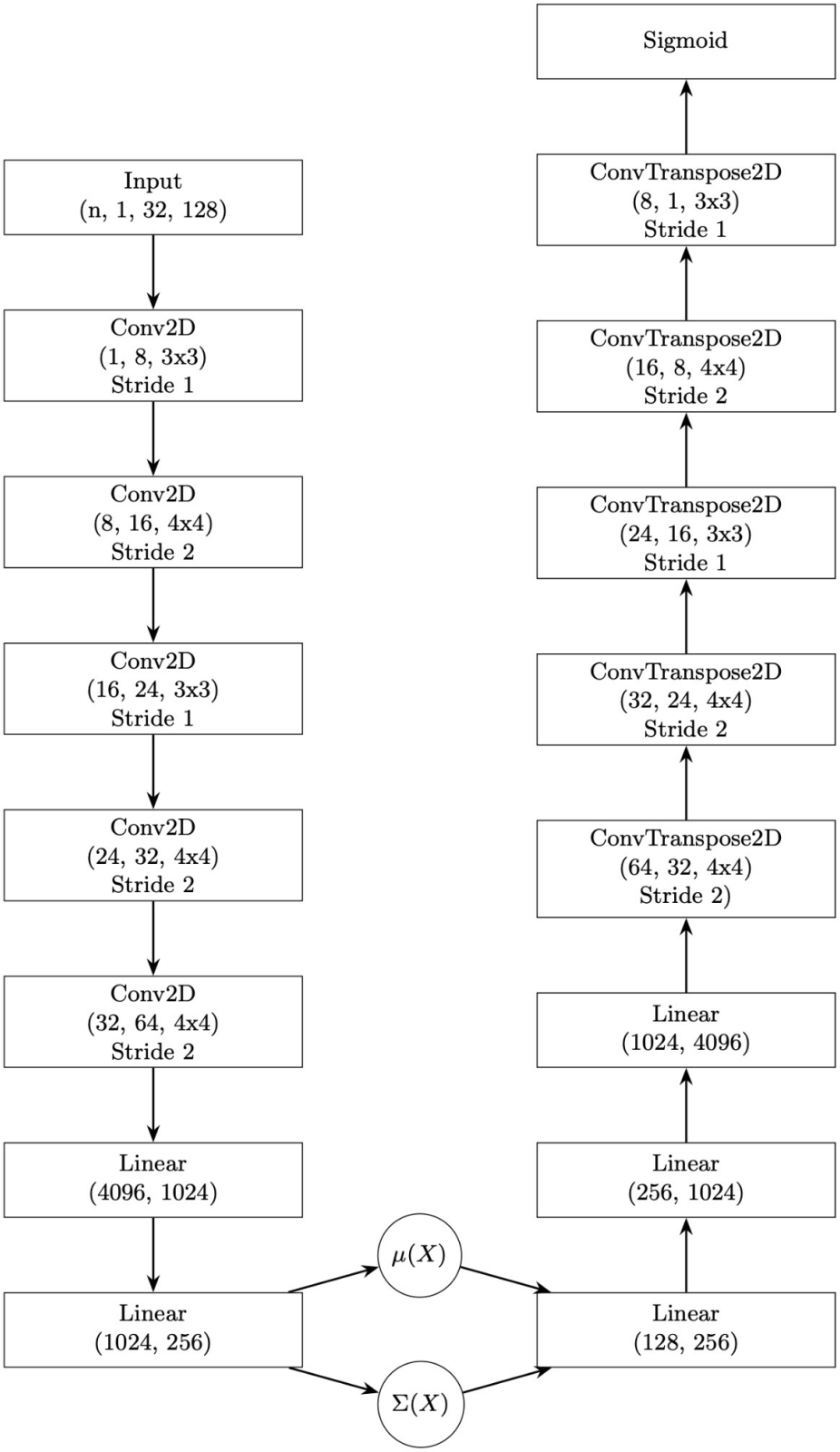

Figure S1 Layer design of the VAE with the encoder on the left and the decoder on the right. Every convolutional layer was followed by a batch normalisation layer. LeakyRELU was used as the activation function for all layer connections. The latent vector has a length of 128.

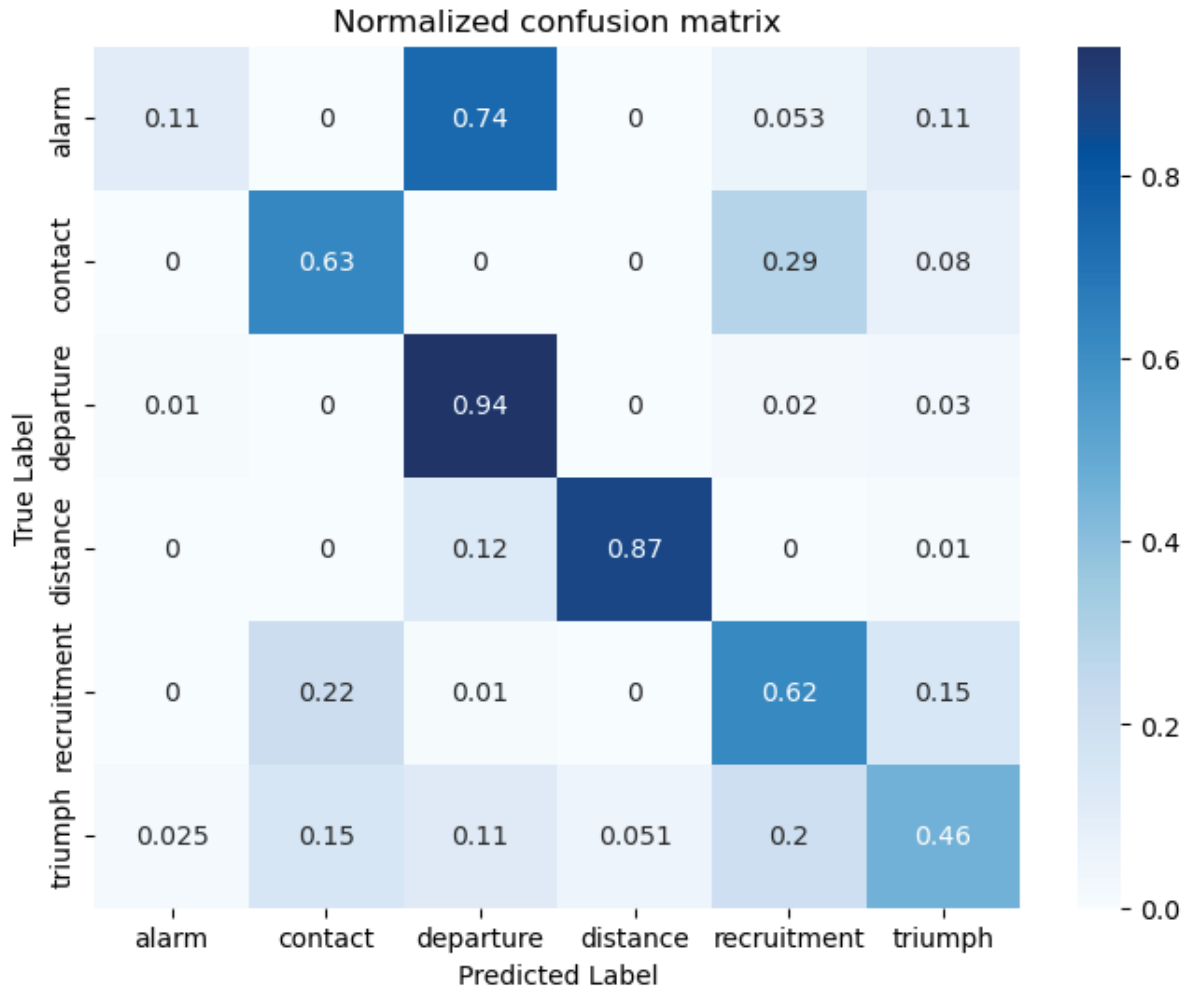

Figure S2 Normalised confusion matrix of the LDA classifier. The model was fitted to a subset of the original dataset with class sizes capped at 500 of which 20 % were set aside for testing. Alarm calls have a low correct prediction rate and are often confused with departure calls. 54 % of the predictions for triumph calls are spread throughout all other categories. Contact and recruitment calls overlap by roughly one quarter in their predictions respectively. Departure and distance calls were largely classified correctly.

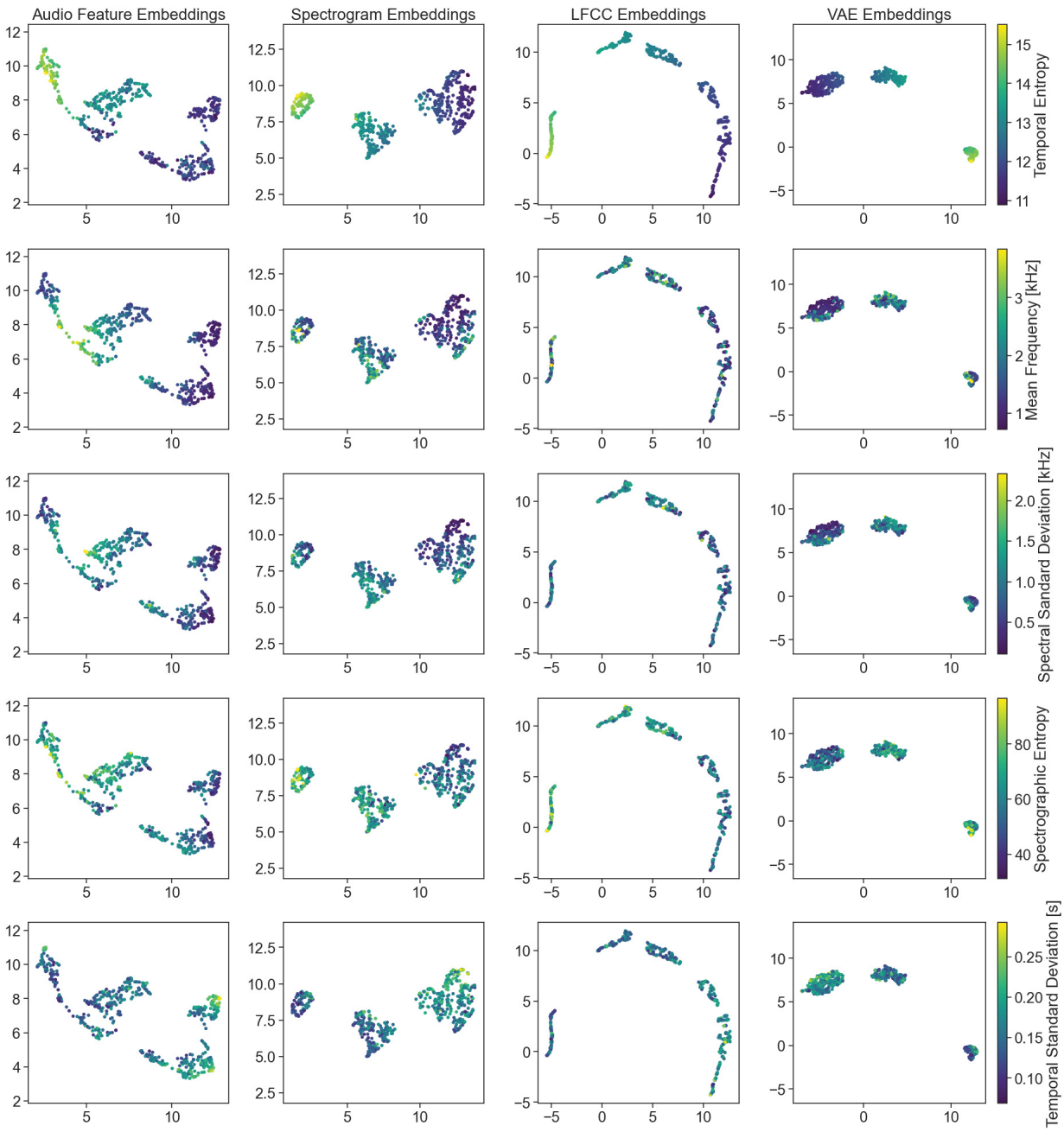

Figure S3 Example projections of every representation, colour-coded for different acoustic features (from top to bottom): temporal entropy, mean frequency in kHz, spectral standard deviation in kHz, spectrographic entropy as the product of temporal and spectral entropy, and temporal standard deviation in seconds.

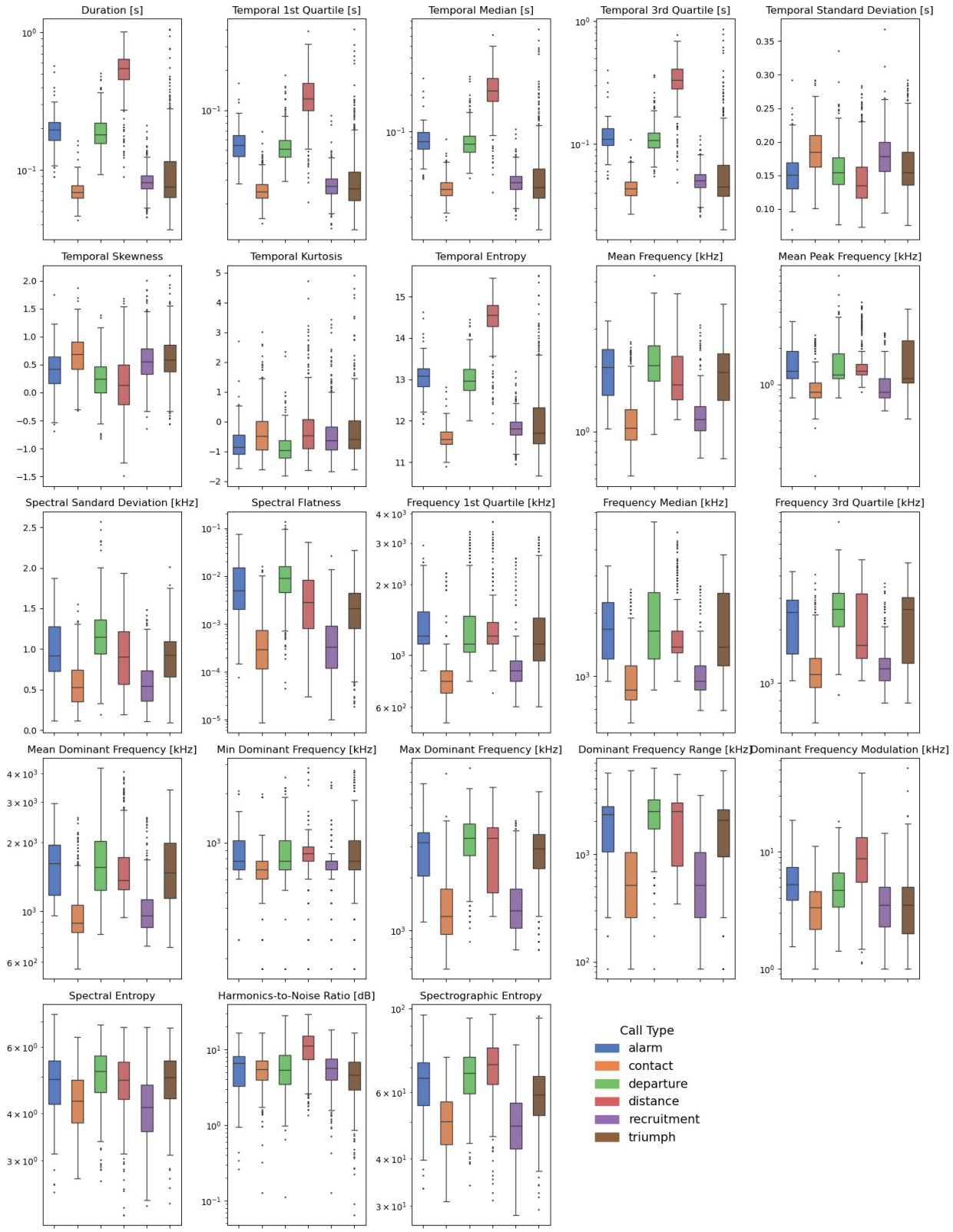

Figure S4 Boxplots for all 23 acoustic features, displaying their distribution per call type.