Decoding Animal Communication

Jayden Dave
Level 4 Project, MPhys Physics
Supervisor: Professor B. Pecjak
Department of Physics, Durham University

Submitted: January 6, 2024

CONTENTS

prepost

1. INTRODUCTION

Non-human animal communication is a field that has fascinated humans for centuries. Parallel to humans, many social animals evolved to communicate with conspecifics by transmitting and receiving complex multimodal signals that convey semantically relevant information [1–3]. These signals employ a diverse range of modalities, often used in combination, encompassing acoustic, visual, tactile, chemical, and electrical. Among these, acoustic vocalisations stand out for their versatility, serving a wide spectrum of purposes, ranging from alerting of impending predators to facilitating social behaviours such as pair bonding. The significance of these vocalisations is underscored by their crucial role in the survival and reproduction of numerous species.

While some taxa appear to exhibit similar forms of communication, a vast array of methods for exchanging information has evolved across the animal kingdom. These variations are driven by both biological and social evolutionary pressures to increase a groups chances of survival within their ecological niche. This evolutionary divergence gave rise to a wide structural diversity of acoustic vocalisations in the animal kingdom [4]. For example, a variety of animals including songbirds, bats and whales can employ a phonological syntax, which includes combining meaningless acoustic units into structured sequences or songs that also appear to be devoid of any context specific meaning (most commonly associated with courtship displays). Alternatively, other species such as chestnut-crowned babblers (*Pomatostomus ruficeps*) can compose meaningful vocalisations from these arbitrary units (analogous to phonemic structuring of words in human languages) that serve functions including coordinating group movement [5]. Other structural patterns include the combination of individually meaningful vocalisations into structures which either reflect the combined meanings of the constituent elements (semantic compositionality) or exhibit entirely unrelated meanings (semantic combinatoriality) [4]. Semantic combinatoriality is mostly seen with species of monkey, an example being putty-nosed monkeys (*Cercopithecus nictitans*) which concatenate calls associated with eagle presences and other disturbances into sequences which promote movement of the group [6]. Manipulating the temporal arrangement of a repeated sound element has also been demonstrated to encode information in species such as the Colobus monkeys (*Colobus guereza* and *Colobus polykomos*) [4, 7]. These monkeys convey categorical information such as the presence of a nearby eagle or leopard by varying the lengths and time intervals of repeated ”roar” sounds .

In order to build our understanding of an animals vocal repertoire, expert ethologists are required to infer the meaning or function of their signals based on observations. These research methods are therefore often very costly and time consuming, as well as raising ethical concerns about disturbing or displacing animals from their natural habitat to a research laboratory [2]. Due to advancements in data collection and accessibility, data driven approaches are becoming increasingly popular to both understand more about animal behaviour as well as for conservation [8]. This addresses the aforementioned concerns as high quality, multimodal data can now be collected automatically and non-invasively through the use of technology such as camera traps, acoustic sensors and bio-loggers. These high quality audio recordings enable comprehensive and impartial computational analysis of animal vocalisations, facilitating investigations into the semantic relevance of acoustic features, particularly those unquantifiable by human senses, such as pitch saliency, spectral mean, and fundamental frequency [3, 9].

Machine learning techniques serve as increasingly powerful tools to investigate animal communication data, offering a diverse set of approaches suited to different research objectives and with varying data and annotation requirements.

A common ML method is supervised learning, where the model aims to predict a human-annotated label from the data, and is regularly employed for classification tasks. In contrast, self-supervised models receive annotation signals from the data itself. A prevalent self-supervised technique involves masking a portion of an input and instructing a neural network to reconstruct the missing data. Self-supervised models are therefore the basis in domains such as natural language processing and includes generative and information restoration models [10].

In the animal communication field, initiatives such as the Earth Species Project (ESP) and Project CETI (Cetacean Translation Initiative) are currently using self-supervised techniques to decode animal communication due to its ability to leverage unannotated data. For instance, ESP's Animal Vocalization Encoder based on Self-supervision (AVES) is a transformer-based audio encoder suited to bioacoustics, allowing for classification and detection tasks [11]. Another self-supervised approached utilised by Project CETI involves leveraging generative models to explore which acoustic features are encoded as meaningful for the sperm whale (*Physeter macrocephalus*) communication system [9]. This innovative method forms the foundation of this work and the utilisation of generative models will be further explored for investigating the vocal repertoires of other species.

2. DEEP LEARNING FOR INVESTIGATING VOCALISATIONS

Deep learning is a powerful tool comprising of multiple computational layers, allowing for higher level representations of data to be obtained in comparison to traditional techniques such as logistic regression [12, 13]. With raw data as input, deep learning neural networks apply non-linear transformations to the data at each layer to learn complex functions. For instance, for classification tasks the network may aim to amplify aspects of the data relevant for discrimination and suppress irrelevancies.

These neural networks can be structured in a variety of forms in order to suit different purposes. A common example is the multilayer perceptron (MLP), comprising of fully connected layers where each neuron/node of adjacent layers are linked, hence being well suited for tasks such as processing numerical data. On the other hand, convolutional neural networks (CNNs) utilise convolutional layers that apply filters to local regions of data, specifically improving pattern recognition for spatially or temporally organised data such as images or audio.

To train networks for their desired function, each layer contains adjustable weights that must be fine tuned to reach optimal values, determined by minimising the network's loss function. This loss function is usually related to the difference (error) between the network's actual and desired outputs- minimising the loss therefore reduces the error. To adjust the weights, the stochastic gradient descent (SGD) process is commonly used. SGD consists of calculating the gradients of the loss function with respect to each weight, indicating the direction they should be adjusted to minimise loss. The weights are then updated by a small amount determined by the chosen learning rate hyperparameter. This iterative process is repeated until the loss function is effectively minimised, with the number of iterations varying with the complexity of the model

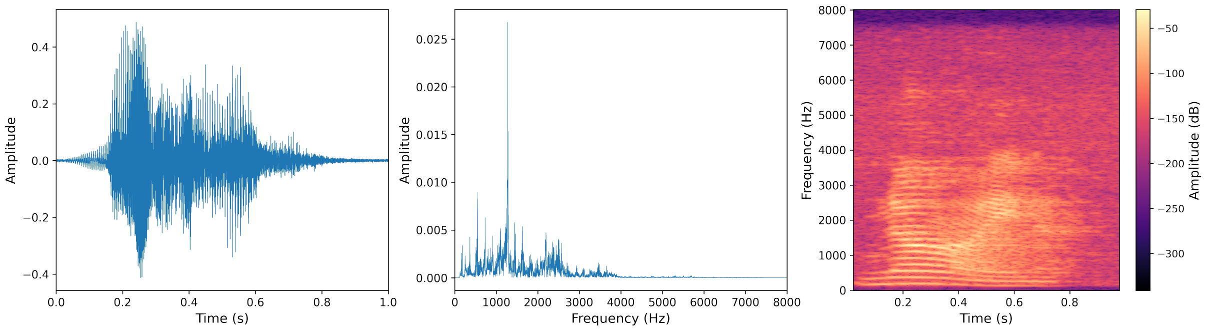

FIG. 1: Acoustic representations of the spoken number ”nine” from the SC09 dataset [14] in the amplitude-time (**left**), amplitude-frequency (**centre**) and frequency-time (**right**) domains. The STFT for the spectrogram (**right**) was computed using SciPy’s spectrogram function, with a Hann window of length 1024 samples and the amplitude log scaled to decibels (dB).

and training data.

2.1. Audio Representations

In the case of animal vocalisations, the audio can be expressed directly as a raw waveform vector, with the numerical value of each sample corresponding to the amplitude at a point in time determined by the sampling rate. Alternatively, the waveforms can be preprocessed into various domains such as amplitude-frequency via a discrete Fourier transform (DFT), which can be computed using a fast Fourier transform (FFT), or a spectrogram representation in time-frequency space utilising a short-time Fourier transform (STFT) [14, 15]. The STFT functions by dividing a signal $x(t)$ into segments and computing the DFT for each one:

$$\text{STFT}(t, f) = \sum_{m=-\infty}^{+\infty} x(t+m)w(m)e^{-ifm}. \tag{1}$$

Here $w(m)$ is a windowing function, increasing the size of the window would include more frequency bins, increasing frequency resolution but reducing the time resolution. Therefore, an appropriate window size much be chosen depending on which resolution is deemed more significant for the specific signal. While the STFT offers a reduction in required computational resources, its reliance on the windowing function introduces assumptions about the characteristics of the signal. When taking a data driven approach to understanding animal communication, it is important to minimise bias in the analysis which could hinder the extraction of important insights. In order to maintain the integrity of the original signal, raw audio waveforms are more appropriate than spectrograms for an unbiased approach [9].

The inherent temporal nature of the raw audio waveforms therefore renders a CNN-based model an natural choice for exploring animal vocalisations through self-supervised, generative deep learning.

3. GENERATIVE ADVERSARIAL NETWORKS

Generative models are ideal candidates for understanding more about animal communication as they must construct their own conclusions in order to generate novel vocalisations without direct access to the training data. These models learn a lower-dimensional representation of the data (referred to as the latent space), which can be probed through methods such as causal inference to reveal the underlying mechanisms that govern acoustic structure [9, 16].

The role of a generative network is to accurately mimic the distribution over the data (the density function), p_{data}, with a model, p_{model}. Therefore, in an ideal case, a sample drawn from p_{model} would be indistinguishable from one drawn from the data distribution itself. However, there are infinitely many possible density functions to approximate the data and for non-trivial problems p_{data} is intractable [16].

One approach to approximate the probability distribution is to model it explicitly whilst constraining the network in some way. For example, autoregressive models enforce a sequentially generated output (e.g., pixel by pixel) by imposing an ordering of the input features. Another explicit approach is to instead model a tractable approximation of p_{data}, examples of these networks are diffusion models, which are trained to denoise a corrupted image, and variational autoencoders (VAEs). VAEs consist of an encoder that translates an input to a lower dimensional representation vector in the latent space, then a decoder which aims to reconstruct the original input from the encoding [17, 18]. The third main approach is to model the distribution implicitly is through a stochastic process that generates data directly, such as generative adversarial networks (GAN) [19].

3.1. The GAN Architecture

Goodfellow et al. (2014) [19] first proposed the generative adversarial network in 2014 as a way of implicitly modelling the p_{data} without the use of approximate inference. The framework consists of two networks in a game theoretic scenario where a generator competes against a discriminator. The generator, $G(\boldsymbol{z};\theta_g)$, and discriminator, $D(\boldsymbol{x};\theta_d)$, are functions represented by either multilayer perceptrons (GAN) or deep convolutional nets (DCGAN) with parameters θ_g and θ_d respectively [20]. The generator takes an input vector $\boldsymbol{z}$ sampled from a multivariate standard normal distribution with prior $p_{\boldsymbol{z}}(\boldsymbol{z})$. These input noise variables are mapped to the data space via $\boldsymbol{x} = G(\boldsymbol{z};\theta_g)$. The discriminator's objective is to determine whether its input data $\boldsymbol{x}$ came from p_{data} rather than p_g (the probability distribution implicitly defined by the generator). The discriminators output layer has a sigmoid activation function and $D(\boldsymbol{x})$ is therefore a scalar representing the probability that $\boldsymbol{x}$ is a real training example.

The training of the generator and discriminator follows a two-player minmax game. Much like a supervised classification task, D is trained to maximise the probability of labelling the generated and real training examples correctly. Meanwhile, G is trained to minimize the likelihood that the discriminator is correct, $\log[1 - D(G(\boldsymbol{z};\theta_g))]$, and therefore achieves this by updating its parameters θ_g to generate more realistic samples. At convergence, the probability distribution defined by the generator would mimic the data distribution, $p_g = p_{\text{data}}$, hence the discriminator would would be unable to differentiate between them, i.e. $D(\boldsymbol{x}) = \frac{1}{2}$ [13, 19].

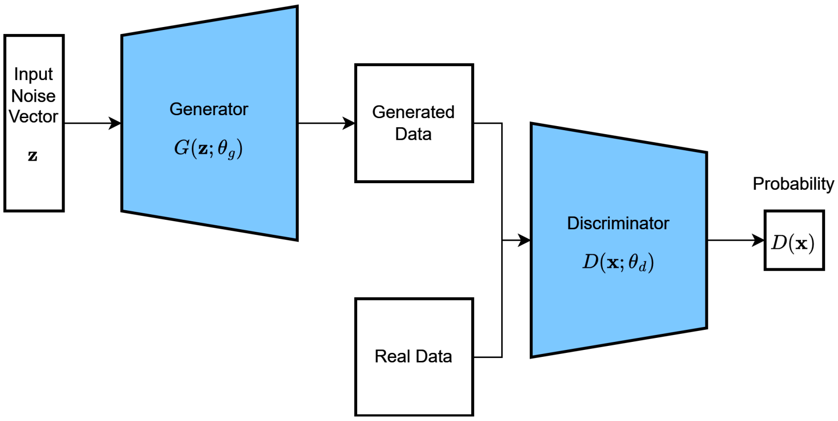

FIG. 2: The original GAN architecture proposed in [19], consisting of a generator which transforms random noise input to synthetic data, and a discriminator that outputs the probability that its input is real.

The minmax game therefore has the value function $V(G,D)$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1-D(G(\boldsymbol{z})))], \tag{2}$$

which would be minimised by G and maximised by D. Minibatch stochastic gradient descent is then utilised to alternately update the parameters θ_d and θ_g, for the discriminator and generator respectively, in order to train the GAN. After sufficient convergence, the discriminator can be discarded and the generator used in isolation to construct a sample that resembles one drawn from p_{data} rather than p_g.

Due to this two-player training method, GANs are particularly adept at generating realistic and high quality data as the generator is forced to continually improve. Generative adversarial networks are also the only architecture where the generator does not see the data directly, unlike the autoencoder for example, where the decoder has access to an encoding (a direct representation of the data) and is trained in a supervised manner. This unsupervised training process allows GANs to learn by imitation and imagination rather than purely replicating training data, giving it an advantage for cognitive modelling over other architectures such as autoencoders, in which the encoder has direct access to the data [19, 21]. For example, when trained on human speech data the network discovers rules that govern the structure and can apply them to generate innovative outputs that were never part of the training data [21]. This is particularly important when applying the lens of decoding animal communication- the GAN architecture has the capability to discover the underlying rules that determine how different animals encode information in their vocalisations.

Unfortunately, GAN training is more difficult in practice as simultaneous gradient descent for both D and G is not guaranteed to reach equilibrium [13]. There are also issues related to balancing out the strength of the generator and discriminator so that one does not overpower the other. In the case that the discriminator becomes too strong, the gradients of the loss function with respect to θ_g become very small, making it difficult for the generator to update its parameters. In such a case, the discriminator must be weakened, a variety of methods can be employed to achieve this such as introducing dropout layers in the discriminator. Dropout layers randomly set the inputs of units from the preceding layer to 0, forcing certain units to take on more or less responsibility and therefore ensuring the network does not become overdependent on specific units [16, 22]. Alternatively, the generator could overpower the discriminator and lead to mode collapse. This occurs when the generator finds a single observation (a mode) that can consistently trick the discriminator, leading to a halt in learning and near-identical samples to be produced.

3.2. Improvements to the GAN training stability

Especially when investigating the complexities of another communication system, it is crucial that the generator's distribution mimics p_{data} as closely as possible, which can be a challenging endeavor due to the previously mentioned training issues. To improve the stability and optimization issues of the GAN, the Wasserstein GAN (WGAN) utilises the Wasserstein (Earth-Mover) distance as the loss function for both the generator and discriminator as opposed to the GAN which is based off the Jensen-Shannon (JS) divergence [19, 23]. The Wasserstein distance is defined as follows:

$$W(p_{\text{data}}, p_g) = \inf_{\gamma \in \Pi(p_{\text{data}}, p_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||], \tag{3}$$

with $\Pi(p_{\text{data}}, p_g)$ denoting the set of all joint distributions $\gamma(x, y)$, where x and y have marginal probability distributions p_{data} and p_g respectively. In essence, $\gamma(x, y)$ is an indication of the "mass" that must be transported from x to y in order to transform p_{data} into p_g and the Wasserstein distance is then the "cost" of the most optimal transport plan. The intractable infimum in (3) can be equated to the following tractable form using the Kantorovich-Rubinstein duality [23, 24]:

$$W(p_{\text{data}}, p_g) = \sup_{||D||_L \leq 1} \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_z}[D(G(\boldsymbol{z})]. \tag{4}$$

The WGAN value function can therefore be defined

$$\min_G \max_{D \in \mathcal{D}} V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[D(G(\boldsymbol{z}))] \tag{5}$$

where $\mathcal{D}$ is the set of 1-Lipschitz continuous functions. A function f is k-Lipschitz continuous if the inequality

$$\frac{|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)|}{|\boldsymbol{x}_1 - \boldsymbol{x}_2|} \leq k \tag{6}$$

is satisfied for any two inputs $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The Wasserstein value function then requires that $D(\boldsymbol{x}) \in [-1, 1]$, hence the sigmoid activation must be removed from the final layer of the discriminator (often referred to as the critic which now outputs a score rather than a probability) [16]. The Lipschitz constraint could be enforced in multiple ways such as clipping the weights of the discriminator [23] or by introducing the gradient penalty term to it's loss function [24]. Since a differentiable function is 1-Lipschitz only if the norm of its gradients have a maximum value of 1 at any point, as per (6), the gradient penalty approach (utilised in the WGAN-GP architecture) considers directly constraining the gradient norm of the discriminator's output. To achieve this, the discriminator's loss function now contains the gradient penalty term with an additional hyperparameter for the penalty coefficient λ [24]:

$$L = \underbrace{\mathbb{E}_{\boldsymbol{z}\sim p_z}[D(G(\boldsymbol{z})] - \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[D(\boldsymbol{x})]}_{\text{Original discriminator loss}} + \underbrace{\lambda\mathbb{E}_{\hat{\boldsymbol{x}}\sim p_{\hat{\boldsymbol{x}}}}[(\|\nabla_{\hat{\boldsymbol{x}}}D(\hat{\boldsymbol{x}})\|_2 - 1)^2]}_{\text{Gradient penalty}}. \tag{7}$$

The gradient penalty term aims to measure the squared distance between the norm of the gradient of the predictions with respect to the input data and 1. However, this calculation is not tractable everywhere. This can be resolved by implicitly defining a distribution $p_{\hat{\boldsymbol{x}}}$ by sampling uniformly along straight lines between pairs of points sampled from p_{data} and p_g. Enforcing the unit gradient norm along these straight lines yields good performance and is an improvement over weight clipping [24]. These changes result in WGAN-GP having an improved stabilization of the training process compared to a standard GAN or DCGAN which will lead to more realistic generations. Another important benefit is that the loss of the generator now correlates with the quality of the samples, which provides much needed clarity when determining a models performance.

3.3. GAN Modifications for Audio Generation

Whilst primarily used for image generation, Generative Adversarial Networks can be applied to other modalities such as text and audio. For images, the DCGAN generator [20] comprises of transposed convolutional layers that iteratively upsample the random noise vector $\boldsymbol{z}$ into a higher resolution image, whilst the discriminator contains convolutional layers that act in an opposite sense to downsample the real or generated images into a singular score. These convolutional layers have two dimensional (5x5) filters that move across the image and record convolutional output. For exploring animal vocalisations with raw audio waveforms, these two dimensional filters must be compressed to longer (25 length), one dimensional filters to suit the one dimensional vector as was introduced for the waveGAN [14] architecture.

3.4. Disentangling Latent Dimensions

While the GAN's ability to produce high-quality data from a random noise vector $\boldsymbol{z}$ is impressive, there are no limitations on how the generator actually incorporates this noise. As a result, it is probable that the noise is utilised in an entangled manner, rendering individual dimensions unrelated to the data's semantic features. Specifically for investigating an unknown field of communication, this is not ideal as determining what the model has deemed semantically meaningful would serve as an important indicator to decipher the main components of a language.

The InfoGAN [25] framework proposes a solution to disentangle these latent representations. The method involves decomposing the generators input vector into two parts: (i) $\boldsymbol{z}$, the standard source of incompressible noise; (ii) $\boldsymbol{c}$, the latent code which aims to encapsulate the salient semantic features of the data.

The latent code can take on a discrete or continuous form. For example, when generating images of digits 0-9 from the MNIST dataset the model generator could be designed to have a discrete code, to represent the numerical identity, and two additional continuous variables that represent the angle and thickness [25**?**].

Through modifications to the architecture, it is possible to discover these latent factors in an unsupervised fashion. The generator now accepts $\boldsymbol{c}$ as an additional input to become $G(\boldsymbol{z}, \boldsymbol{c})$. Additional constraints must also be applied as a standard GAN could simply chose to ignore the latent code, hence finding an undesirable solution that satisfies $p_g(\boldsymbol{x}|\boldsymbol{c}) = p_g(\boldsymbol{x})$. The solution lies within information theory, where the model should be regularised to promote high mutual information between the generator distribution $G(\boldsymbol{z}, \boldsymbol{c})$ and the latent code $\boldsymbol{c}$. A large mutual information $I(\boldsymbol{c}; G(\boldsymbol{z}, \boldsymbol{c}))$ signifies that there is a high amount of information learned about $\boldsymbol{c}$, given knowledge of $G(\boldsymbol{z}, \boldsymbol{c})$. This can be expressed in terms of entropy:

$$\begin{aligned} I(\boldsymbol{c}; G(\boldsymbol{z}, \boldsymbol{c})) &= H(\boldsymbol{c}) - H(\boldsymbol{c}|G(\boldsymbol{z}, \boldsymbol{c})) & (8) \\ &= H(\boldsymbol{c}) + \mathbb{E}_{\boldsymbol{x}\sim G(\boldsymbol{z},\boldsymbol{c})}[\mathbb{E}_{\boldsymbol{c}'\sim P(\boldsymbol{c}|\boldsymbol{x})}[\log P(\boldsymbol{c}'|\boldsymbol{x})]] & (9) \end{aligned}$$

where $H(\boldsymbol{c})$ is the entropy of the latent code and $H(\boldsymbol{c}|G(\boldsymbol{z}, \boldsymbol{c}))$ is the conditional entropy of $\boldsymbol{c}$ after observing the generator output. Therefore, maximising the mutual information correlates to minimising $H(\boldsymbol{c}|G(\boldsymbol{z}, \boldsymbol{c}))$- decreasing the uncertainty in the latent code after observing an output from G. However, for any given $\boldsymbol{x} \sim p_g(\boldsymbol{x})$ access to the posterior $P(\boldsymbol{c}|\boldsymbol{x})$ is required (Eq. (9)) which cannot be calculated directly.

The InfoGAN framework circumvents this issue by defining an auxiliary distribution $Q(\boldsymbol{c}|\boldsymbol{x})$ to approximate $P(\boldsymbol{c}|\boldsymbol{x})$, leading to a lower bound of the mutual information utilising the Variational Information Maximization technique [26]:

$$L_I(G, Q) = \mathbb{E}_{\boldsymbol{c}\sim P(\boldsymbol{c}),\boldsymbol{x}\sim G(\boldsymbol{z},\boldsymbol{c})}[\log Q(\boldsymbol{c}|\boldsymbol{x})] + H(\boldsymbol{c}) \leq I(\boldsymbol{c}; G(\boldsymbol{z}, \boldsymbol{c})). \tag{10}$$

The value function for the InfoGAN framework is therefore defined with the addition of the regularization term, with hyperparameter λ, to maximise the lower bound of the mutual information:

$$\min_{G,Q} \max_{D} V_{\text{InfoGAN}}(D, G, Q) = V(D, G) - \lambda L_I(G, Q). \tag{11}$$

The auxiliary distribution Q is parameterized as a neural network that can (in the case of InfoGAN) share all convolutional layers with the discriminator D, with a differing output layer activation function depending on whether the latent codes are categorical or continuous. When utilising Wasserstein loss, the discriminator is a 1-Lipschitz constrained function which may not be desirable for for the auxiliary network. For this reason, the parameters θ_d and θ_q for the discriminator and Q networks respectively are separate in some architectures [21].

Another approach to modelling features is to utilise binary code for $\boldsymbol{c}$ (e.g. [1,0,0,1,0]) as in the fiwGAN architecture [21]. This has been shown to have advantages for modelling classification as well as featural learning of phonetic and phonological representations in human speech. The fiwGAN architecture combines the disentangles representations from InfoGAN with the audio compatibility of WaveGAN, while also employing the Wasserstein loss function for improved training stability. The architecture has also been successfully trained on sperm whale codas (vocalisations) and learned meaningful properties such as the inter-click intervals and number of clicks in a coda [9].

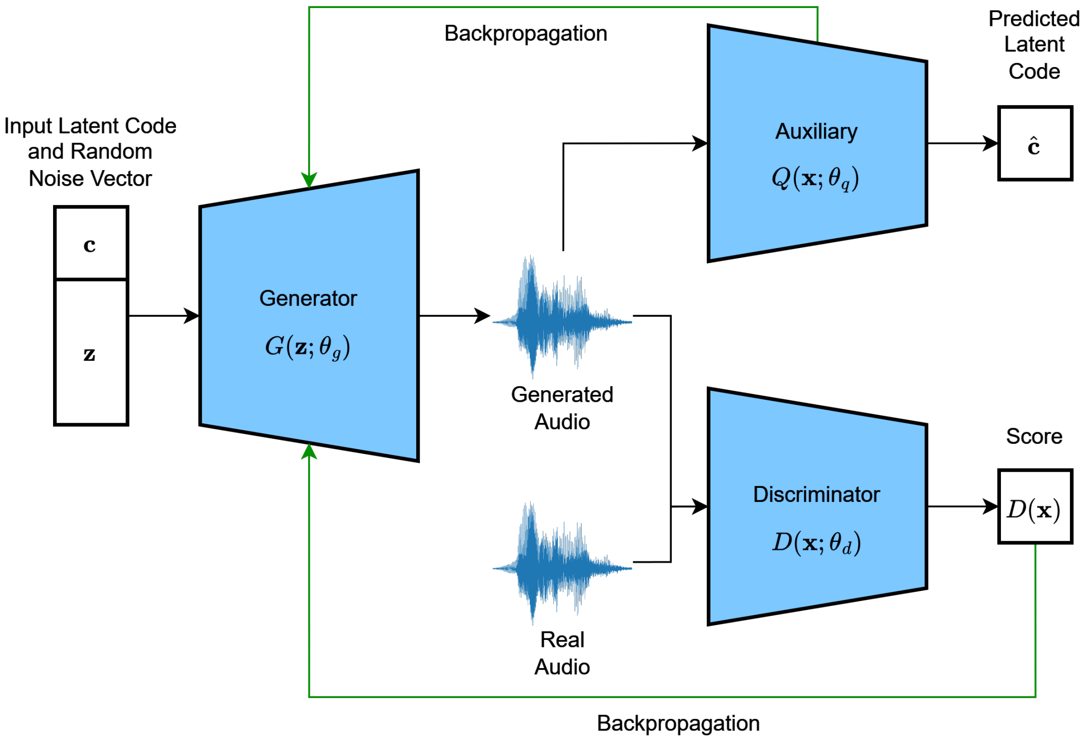

FIG. 3: An overview of the architecture for disentangled representation models such as InfoGAN [25] and fiwGAN [21]. The weights of the generator, auxiliary and discriminator networks are updated through minibatch stochastic gradient descent and backpropagation in relation to their specified loss functions.

3.5. Causal Disentanglement with Extreme Values (CDEV) Technique

Although the featural encoding c is trained in an unsupervised fashion, causal inference techniques can be employed to gain insight into how it corresponds to any observable acoustic properties [9]. As this latent encoding space is small in comparison to the incompressible noise (e.g. vectors of 5 and 95 dimensions for c and z respectively), the generator is forced to place higher emphasis on each bit in c during training and therefore the latent code should convey properties that the model considers salient.

After decoupling the generator, an input vector can be constructed by concatenating the randomly sampled incompressible noise $z \sim \text{Unif}(0, 1)$ with a manually set latent code c. By systematically manipulating c within the input vectors, diverse samples can be generated which can be probed to explore the relationship between the latent code and the acoustic properties. The causal disentanglement with extreme values (CDEV) approach manually sets individual bits in c to a desired value (the dosage), whilst setting the others to a baseline value. This dosage can be set as high as possible, but comes at the price of increasing noise on the generated data. Setting a bit to an extreme value forces disentanglement as the the generator must put more emphasis on that specific bit, allowing the encoded properties to be observed.

4. DATA GENERATION

Data generation was achieved utilising the fiwGAN architecture [21], with incompressible noise vector $\boldsymbol{z}$ and featural latent code $\boldsymbol{c}$ (of lengths 95 and 5 respectively) as the input to the generator. An overview of the architecture's constituent generator, discriminator and auxiliary networks is shown in Table I. With five 1D transposed convolutional layers for up-sampling, the generator outputs 16,384 data points (corresponding to approximately 1 second of audio at a 16kHz sampling rate) which are then used as input for the auxiliary network and discriminator, alongside the training data.

(a) Generator

Operation	Output Shape
Input $[\boldsymbol{z};\boldsymbol{c}]$	(n, 100)
Dense	(n, 16384)
Reshape	(n, 16, 1024)
ReLU	(n, 16, 1024)
Trans Conv1D	(n, 64, 512)
ReLU	(n, 64, 512)
Trans Conv1D	(n, 256, 256)
ReLU	(n, 256, 256)
Trans Conv1D	(n, 1024, 128)
ReLU	(n, 1024, 128)
Trans Conv1D	(n, 4096, 64)
ReLU	(n, 4096, 64)
Trans Conv1D	(n, 16384, 1)
Tanh	(n, 16384, 1)

(b) Discriminator

Operation	Output Shape
Input $\boldsymbol{x}$ or $G(\boldsymbol{z}, \boldsymbol{c})$	(n, 16384, 1)
Conv1D	(n, 4096, 64)
Leaky ReLU	(n, 4096, 64)
Phase Shuffle	(n, 4096, 64)
Conv1D	(n, 1024, 128)
Leaky ReLU	(n, 1024, 128)
Phase Shuffle	(n, 1024, 128)
Conv1D	(n, 256, 256)
Leaky ReLU	(n, 256, 256)
Phase Shuffle	(n, 256, 256)
Conv1D	(n, 64, 512)
Leaky ReLU	(n, 64, 512)
Phase Shuffle	(n, 64, 512)
Conv1D	(n, 16, 1024)
Leaky ReLU	(n, 16, 1024)
Flatten	(n, 16384)
Dense	(n, 1)

(c) Auxiliary

Operation	Output Shape
Input $G(\boldsymbol{z}, \boldsymbol{c})$	(n, 16384, 1)
Conv1D	(n, 4096, 64)
Leaky ReLU	(n, 4096, 64)
Conv1D	(n, 1024, 128)
Leaky ReLU	(n, 1024, 128)
Conv1D	(n, 256, 256)
Leaky ReLU	(n, 256, 256)
Conv1D	(n, 64, 512)
Leaky ReLU	(n, 64, 512)
Conv1D	(n, 16, 1024)
Leaky ReLU	(n, 16, 1024)
Flatten	(n, 16384)
Dense	(n, 5)

TABLE I: The fiwGAN architecture for the **(a)** generator, **(b)** discriminator and **(c)** auxiliary networks, where *n* is the batch size. All Conv1D and Trans Conv1D layers have a stride length of 4 and kernel size of 25, while the Leaky ReLU and Phase Shuffle layers have parameters 0.2 and 2 respectively.

4.1. Leveraging Human Speech Analysis

Before venturing into the complexities of non-human communication systems, utilising generative models and causal inference techniques to investigate human vocalisations offers several advantages. The first advantage is that both the acoustic quality (e.g. noise levels) and the naturalness (including intelligibility) of the samples can be qualitatively assessed by human perception. This immediate feedback can indicate whether the model requires improvement, whereas gauging the intelligibility of non-human vocalisations often requires expert analysis

[2]. Another reason for applying these methods on human language is that there exists extensive established research in the field, facilitating comparison to animal vocalisations, as well as a means to validate the approach.

For these reasons, initial assessment of the fiwGAN architecture and CDEV methodology was carried out using English speech samples of five different digits from the *Speech Commands Zero Through Nine* (SC09) dataset [14]. This dataset consists of 1 second recordings of the spoken digits ”zero” through ”nine” from many different speakers and captures a variety of acoustic properties that can be investigated.

To ensure compatibility with the model architecture, the training data was down-sampled to 16kHz (from 22kHz) and zero-padded to reach 16,384 data points.

4.2. Model Training

The model was implemented with TensorFlow and Keras in Python and trained numerous times whilst systematically adjusting learning rates, training data size and other parameters until the generator could produce intelligible samples. The generated data utilised in this investigation was produced from a training loop where for each generator and auxiliary update, the discriminator underwent 5 updates. Both G and D were trained with the Adam optimizer, whereas Q training utilised the RMSProp algorithm- a consistent learning rate of 0.00005 was set for all optimizers. 1280 samples of data (containing an equal number of the randomly chosen spoken numbers ”nine”, ”eight”, ”seven”, ”three” and ”one”) were used to train the model, with a batch size of 64, for 15,000 epochs- totalling $\sim$ 5 days of training on a NVIDIA Tesla V100 32GB GPU. Checkpoints were taken every 1000 epochs and best performance was found at 5000 epochs of training, this was the model chosen.

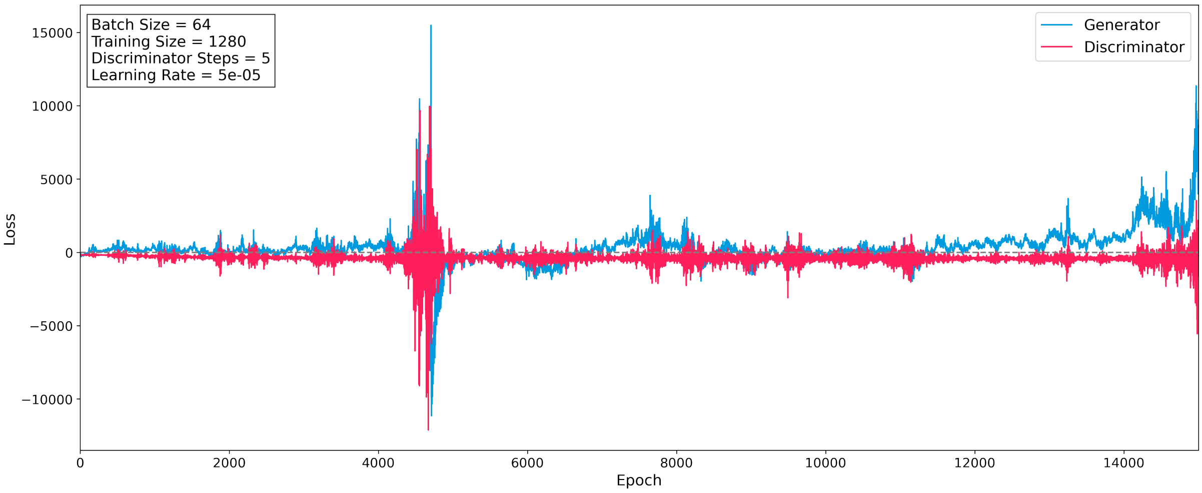

FIG. 4: Losses for the fiwGAN generator and discriminator throughout training. The discriminator loss includes the effect of the Wasserstein loss and the gradient penalty term with coefficient $\lambda = 10$. Both loss functions fluctuate around zero as the generator improves to trick the discriminator.

5. INVESTIGATING THE GENERATED SAMPLES

After decoupling the generator, the CDEV technique can be applied to generate speech whilst disentangling the latent variables. To achieve this, a baseline dose of $t' = [0, 0, 0, 0, 0]$ is established for the featural encoding, with individual bits systematically set to values ranging from -1 to 13, excessively noisy samples were generated beyond this value. In order to mitigate any variability stemming from the incompressible noise input, 250 samples were generated at each instance and their acoustic properties calculated and averaged.

5.1. Fundamental Frequency

An important property in human vocalisations is the fundamental frequency (F0), which can convey acoustic cues such as pitch, vowel identity and intonation;it can also provide information about the speaker's age, gender and emotional state [27].

F0 was calculated for each generated sample by isolating the frequency with the largest amplitude from the FFT.

The average fundamental frequency per dosage, t, for each bit is visualised in Figure 5. The results indicate that both bits 3 and 4 are influential to F0, with bit 3 causing a rapid increase at higher doses. However, the large standard deviation associated with bit 3 at these doses indicates the presence of noise or high frequency artifacts.

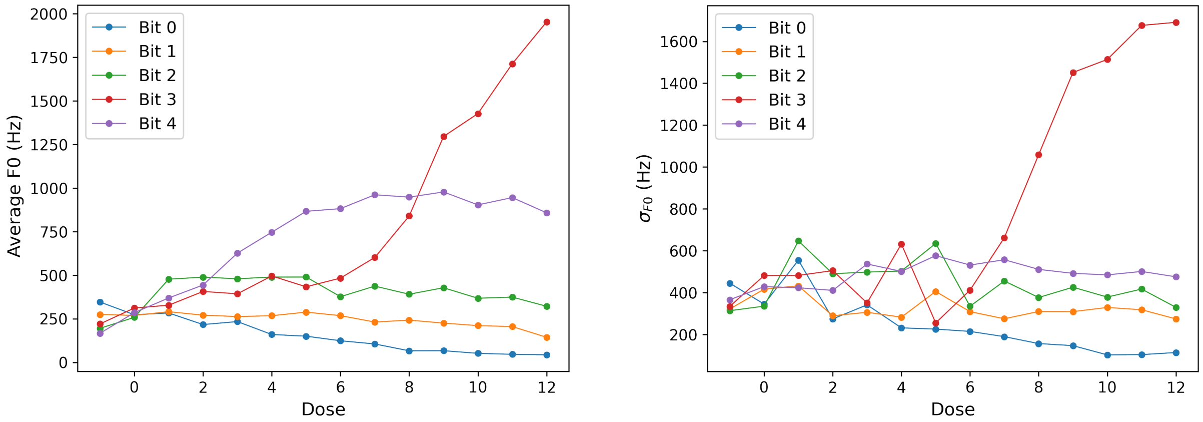

FIG. 5: The average fundamental frequency (F0) (**left**) and standard deviation (**right**) with varying dosage for each bit.

In either case, the encoding of F0 (or a property with relation to F0) in bits 3 and 4 demonstrate that the model has placed high importance on it and is therefore in agreement with established knowledge [9, 27]. This same procedure, along with other indicators, can thus be utilised to investigate the importance of F0 and other acoustic properties in the field of non-human animal communication.

6. CONCLUSIONS AND FUTURE WORK

In conclusion, a form of generative adversarial network (fiwGAN) was utilised to generate samples of human speech. These samples were probed using the causal disentanglement with extreme values (CDEV) technique to determine which acoustic properties the model considered meaningful. Through this causal inference method, it was determined that bits 3 and 4 encode some form of relation to the fundamental frequency, demonstrating it's importance for human speech.

Future work could see how these techniques could be used to explore other acoustic properties that are meaningful for non-human animal communication systems.

REFERENCES

[1] R. M. Seyfarth and D. L. Cheney, "Signalers and receivers in animal communication," Annual Review of Psychology, vol. 54, no. 1, pp. 145–173, 2003. PMID: 12359915.

[2] C. Rutz, M. Bronstein, A. Raskin, S. C. Vernes, K. Zacarian, and D. E. Blasi, "Using machine learning to decode animal communication," Science, vol. 381, no. 6654, pp. 152–155, 2023.

[3] J. E. Elie and F. E. Theunissen, "The vocal repertoire of the domesticated zebra finch: a data-driven approach to decipher the information-bearing acoustic features of communication signals," Animal Cognition, vol. 19, pp. 285–315, 2016.

[4] S. Engesser and S. W. Townsend, "Combinatoriality in the vocal systems of nonhuman animals," WIREs Cognitive Science, vol. 10, no. 4, p. e1493, 2019.

[5] S. Engesser, J. M. S. Crane, J. L. Savage, A. F. Russell, and S. W. Townsend, "Experimental evidence for phonemic contrasts in a nonhuman vocal system," PLOS Biology, vol. 13, pp. 1–16, 06 2015.

[6] K. Arnold and K. Zuberbühler, "Call combinations in monkeys: Compositional or idiomatic expressions?," Brain and Language, vol. 120, no. 3, pp. 303–309, 2012.

[7] A. M. Schel, A. Candiotti, and K. Zuberbühler, "Predator-deterring alarm call sequences in guereza colobus monkeys are meaningful to conspecifics," Animal Behaviour, vol. 80, no. 5, pp. 799–808, 2010.

[8] D. Tuia, B. Kellenberger, S. Beery, B. R. Costelloe, S. Zuffi, B. Risse, A. Mathis, M. W. Mathis, F. van Langevelde, T. Burghardt, R. Kays, H. Klinck, M. Wikelski, I. D. Couzin, G. van Horn, M. C. Crofoot, C. V. Stewart, and T. Berger-Wolf, "Perspectives in machine learning for wildlife conservation," Nat Commun, vol. 13, no. 1, p. 792, 2022.

[9] G. Beguš, A. Leban, and S. Gero, "Approaching an unknown communication system by latent space exploration and causal inference," 2023.

[10] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum, "A cookbook of self-supervised learning," 2023.

[11] M. Hagiwara, "Aves: Animal vocalization encoder based on self-supervision," 2022.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015.

[13] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[14] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," 2019.

[15] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, Discrete-Time Signal Processing. Prentice-hall Englewood Cliffs, second ed., 1999.

[16] D. Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play. O'Reilly Media, Inc., 2 ed., 2023.

[17] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," 2023.

[18] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.

[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[20] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016.

[21] G. Beguš, "Ciwgan and fiwgan: Encoding information in acoustic data to model lexical learning with generative adversarial networks," Neural Networks, vol. 139, pp. 305–325, 2021.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, no. 56, pp. 1929–1958, 2014.

[23] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," 2017.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," 2017.

[25] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," 2016.

[26] D. Barber and F. V. Agakov, "The im algorithm: a variational approach to information maximization," in Neural Information Processing Systems, 2003.

[27] J. H. Lee and L. E. Humes, "Effect of fundamental-frequency and sentence-onset differences on speech-identification performance of young and older adults in a competing-talker background," The Journal of the Acoustical Society of America, vol. 132, pp. 1700–1717, Sept. 2012.