GUIDED DISENTANGLED REPRESENTATION LEARNING FROM AUDIO DATA FOR TRANSFER LEARNING

A Thesis submitted by

Kazi Nazmul Haque
BURP, MIT

For the award of

Doctor of Philosophy

2024

ABSTRACT

In the field of machine learning, disentangled representation learning seeks to map high-dimensional data into a low-dimensional space where the underlying variational factors are both disentangled and easily separable. This thesis investigates the application of such representations, derived from unlabelled data to tasks where only limited labelled data is available. Specifically, I explore the domain of audio modelling, where the absence of supervision in learning representations from unlabelled data often results in representations that may not be optimally useful for downstream tasks, leading to potential resource wastage. To address this issue, I introduce the Guided Generative Adversarial Neural Network (GGAN), a novel model that utilises a modest amount of labelled data to guide the learning of relevant disentangled representations from a larger corpus of unlabelled data. While the representation learned through GGAN proves beneficial for the task at hand, its generalisation capabilities are limited, restricting the model's application to tasks similar to or closely related to the original one. To overcome this limitation, I propose a second model, the Guided Generative Adversarial Autoencoder (GAAE), which not only learns representations tailored to a specific downstream task but also captures the general attributes of the data, thereby being independent of the particular task. Both GGAN and GAAE are founded on the Generative Adversarial Network (GAN) architecture, leveraging the audio generalisation prowess of GANs for representation learning. Nevertheless, the models eschew working with 1D raw audio waveforms directly, instead utilising 2D spectrograms, a practice that recent research suggests may curtail the models' ultimate performance capabilities, representing a significant gap in the literature. This thesis confronts this issue head-on. Convolutional Neural Networks (CNNs), forming the structural backbone of both GGAN and GAAE, have historically faced challenges in generating raw audio waveforms via adversarial training. A foundational step in surmounting this hurdle involves a thorough examination of CNNs' ability to model raw audio waveforms, such as classification tasks. Moving strategically in this direction, I have proposed two cosine filter-based CNN models: the Cosine Convolution Neural Network (CosCovNN) and the Vector Quantised Cosine Convolutional Neural Network with Memory (VQCCM). These models have not only outclassed traditional CNN architectures but have also set a new benchmark in the field of audio classification.

CERTIFICATION OF THESIS

I Kazi Nazmul Haque declare that the PhD Thesis entitled *Guided Disentangled Representation Learning from Audio Data for Transfer Learning* is not more than 100,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references, and footnotes.

This Thesis is the work of Kazi Nazmul Haque except where otherwise acknowledged, with the majority of the contribution to the papers presented as a Thesis by Publication undertaken by the student. The work is original and has not previously been submitted for any other award, except where acknowledged.

Date: 23/01/2024

Endorsed by:

Prof. Rajib Rana
Principal Supervisor

Prof. Ji Zhang
Associate Supervisor

Student and supervisors' signatures of endorsement are held at the University.

STATEMENT OF CONTRIBUTION

The papers produced from this doctoral Thesis are a joint contribution of the student and supervisory team. However, most of the work is completed by the student. The details of the contribution are as follows,

Paper 1:

Kazi Nazmul Haque, Rajib Rana, Jiajun Liu, John H. L. Hansen, Nicholas Cummins, Carlos Busso and Björn W. Schuller. "**Guided Generative Adversarial Neural Network for Representation Learning and Audio Generation Using Fewer Labelled Audio Data**," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2575-2590, 2021, doi: 0.1109/TASLP.2021.3098764.

Student contributed 75% to this paper. Collectively Rajib Rana, Jiajun Liu, John H. L. Hansen, Nicholas Cummins, Carlos Busso and Björn W. Schuller contributed the remainder.

Paper 2:

Kazi Nazmul Haque, Rajib Rana, and Björn W. Schuller, "**High-Fidelity Audio Generation and Representation Learning with Guided Adversarial Autoencoder**," in *IEEE Access*, vol. 8, pp. 223509-223528, 2020, doi: 0.1109/ACCESS.2020.3040797.

Student contributed 80% to this paper. Collectively Rajib Rana, and Björn W. Schuller contributed the remainder.

Paper 3:

Kazi Nazmul Haque, Rajib Rana, and Björn W. Schuller, "**Raw Audio Classification with Cosine Convolutional Neural Network (CosCovNN)"** submitted to *IEEE Access, 2024*.

Student contributed 80% to this paper. Collectively Rajib Rana, and Björn W. Schuller contributed the remainder.

ACKNOWLEDGEMENTS

I wish to express my heartfelt gratitude to the individuals and institutions who have played pivotal roles in my journey during this PhD:

Dr. Rajib Rana, my principal supervisor, has been not only a guiding light but also a source of life wisdom. His recognition of my potential and unwavering support have been instrumental in shaping my academic and professional career. I extend my sincerest thanks to my associate supervisors, Dr. Zi Zang, and esteemed researchers, including Jiajun Liu, John H. L. Hansen, Nicholas Cummins, Carlos Busso, and Björn W. Schuller. Their feedback and insights have significantly contributed to the refinement of my research and writing, especially in the context of journal publications.

Financial support provided by the Advance Queensland PhD, USQ International Fees Research Scholarship, and the Australian Government Research Training Program Scholarship has been invaluable in facilitating my research endeavours. The University of Southern Queensland has provided me with a platform to complete this PhD. I appreciate their commitment to fostering academic growth and providing the necessary resources.

To my parents, whose sacrifices and dedication have paved the way for my education at esteemed institutions, I offer my deepest gratitude. Their unwavering support and love have brought me to this juncture. I'm equally thankful for the affection and encouragement of my younger sister. Lastly, my profound appreciation goes to my wife, who has stood by my side for the last 16 years, offering unwavering support throughout life's challenges. Her hard work has allowed me to concentrate on my studies, and her presence has been a constant source of strength.

I am truly grateful to each of you for your contributions, unwavering support, and belief in my academic and personal journey. Your encouragement has been the driving force behind my achievements, and I will forever cherish your kindness and guidance.

DEDICATION

I dedicate this thesis to the unwavering love, support, and sacrifices of my family, who have been my pillars of strength throughout this journey:

To my beloved parents, who have tirelessly worked and sacrificed to provide me with opportunities and a bright future. Your love and unwavering belief in me have been my motivation.

In loving memory of my late Nanu, Master Nanu and Apa, whose wisdom and guidance have left a lasting imprint on my life. I carry your legacy with me in all that I do.

To my incredibly beautiful wife, whose love, patience, and understanding have been my anchor during the challenges of this journey. Your unwavering support has made all the difference.

To my precious son, whose contagious smile and boundless love have brought immeasurable joy to my life. Your laughter is my daily inspiration.

I also dedicate this work to the Almighty, whose grace and blessings have illuminated my path and granted me the strength to overcome obstacles.

This accomplishment is a testament to the love and support of my family and the divine guidance of Allah. I dedicate this thesis with heartfelt gratitude and love.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1. Background

Feature extraction and data representation are essential components that significantly impact the performance of machine learning models. How data is represented significantly affects how well models can recognise patterns, make predictions, and apply these predictions to new data [1]. In the past, machine learning researchers heavily focused on creating models to improve data representation techniques [2]. However, with the advent of deep learning, the focus has shifted. Deep learning can independently learn representations from raw data, eliminating the need for manually crafted features, which has contributed to its success, especially in supervised learning tasks [3].

Nonetheless, deep learning based supervised algorithms rely heavily on extensive labelled datasets, which can be expensive, time-consuming, and sometimes impossible to obtain. In the present age of the Internet of Things (IoT), there is an abundance of data accessible on the Internet, with organisations continuously generating substantial data volumes. However, supervised learning systems encounter challenges in harnessing these extensive datasets due to their lack of labelling [4].

Unsupervised representation learning, a subfield of unsupervised learning, offers a promising solution to these challenges. It involves creating a machine learning model that can learn data representations without the need for labelled data. During this process, the model transforms high-dimensional data into a lower-dimensional representation space, where the inherent data features are separated, making them easier to separate using basic machine learning models. The process of learning these separated features is referred to as the "disentangled representation learning” [5, 6].

When the model can learn disentangled representation from any unlabelled data, this learning can be applied to related supervised tasks where limited labelled data is available. This approach, known as transfer learning, allows us to utilise the vast amount of unlabelled big data effectively [7]. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, for instance, consider an emotion classification task based on audio data, where only a small amount of labelled dataset is available. Training a machine learning algorithm on such

a limited dataset poses significant challenges in accurately learning emotion-related information. However, with an abundance of unlabelled audio data, such as related open-sourced YouTube audio content, where emotion-related information is embedded and entangled, a model capable of learning disentangled representations from this vast dataset can effectively learn to disentangle emotion-related information. When this trained model is used to extract features from that limited labelled dataset, it is expected to substantially improve the emotion classification accuracy.

Unsupervised representation learning looks intriguing as it can utilise an enormous amount of unlabelled data. However, it's important to note that learning representations in an unsupervised manner doesn't necessarily ensure their post-use case scenario [8]. Recent research by Locatello et al. has demonstrated that achieving entirely unsupervised representation learning is not feasible without some form of supervised signal [9].

In this thesis, I align with the approach of unsupervised representation learning using supervised signals within the domain of audio processing. My research was initially motivated by existing literature, where researchers [10-12] have introduced models based on Generative Adversarial Neural Networks (GANs) [13] to guide unsupervised representation learning with the supervision of a limited amount of labelled data. A GAN typically comprises two neural networks: the Generator and the Discriminator. These networks undergo training through an alternating minimax-game optimisation process. In this training setup, the Discriminator's objective is to differentiate between real samples obtained from a data distribution and counterfeit samples generated by the Generator. Conversely, the Generator aims to deceive the Discriminator by generating samples that closely resemble real ones. During this process of generating real-like samples Generator learns to capture disentangled representation in its latent variable [14-17].

The effectiveness of GAN-based models in representation learning hinges on the quality of the generated samples. In the domain of audio generation, GAN-based models encounter difficulties when attempting to generate raw audio waveforms directly. Therefore, researchers have directed their efforts towards strategies for generating low-dimensional acoustic features or representations, such as audio spectrograms (2D image-like representation of audio), instead of attempting to generate the raw waveform itself. Subsequently, these spectrograms are transformed back into the audio format [18, 19]. In my research, I follow this direction by converting

audio data into 2D image-like log-magnitude spectrograms using the short-time Fourier Transform. The generated log-magnitude spectrograms of the models are then transformed into audio using the PGHI algorithm [20]. In the rest of the research, I refer to the log-magnitude spectrogram as the spectrogram.

In this thesis, I introduced a novel model called the Guided Generative Adversarial Neural Network (GGAN). GGAN excels at generating high-quality spectrograms and acquiring task-specific disentangled representations from unlabelled data with guidance from a limited, related labelled dataset. Through extensive experimentation, I have demonstrated that GGAN enhances audio classification tasks, even when provided with a very small subset (approximately 5%) of data while training along with a substantial amount of related unlabelled data. This is a joint training approach, where both the labelled and unlabelled datasets are used together during the training. Nevertheless, it's essential to note that the representations learned with GGAN are highly task-specific and may not generalise well for unrelated tasks.

In many cases, it is desirable to learn representation in a manner so that it can be used for any particular downstream task as well as can be used for any future tasks independent of the downstream task at hand [21]. It is a challenging problem for GGAN to learn both generalised and guided representations in the same latent space. Therefore, I also address this issue in this thesis by proposing a novel autoencoder-based model named Guided Adversarial Autoencoder (GAAE). GAAE can generate high-quality audio samples that capture different modes of the training data, guided by a small, labelled dataset. Through the power of audio generation, GAAE can learn guided representations tailored to the labelled dataset used during training, as well as general representations that are not tied to any specific task.

Here, both GGAN and GAEE work with the spectrogram of the audio as GAN-based models struggle with the complex audio waveform generation as it requires modelling higher-order temporal scale [19]. However, Ravanelli and Bengio, argued in their research [22] that the optimality of the hand-crafted features such as spectrogram is not guaranteed as they are designed with perceptual evidence only. Moreover, models working with these manually extracted features might not be able to utilise the data to its full potential [23]. Therefore, using spectrograms as input to GGAN and GAAE models may limit their full potential. Since Convolutional Neural Networks (CNNs) serve as the core components of the proposed models and CNNs encounter

challenges when it comes to directly modelling audio waveforms, there is a need to enhance the CNN model's capabilities for raw audio waveform generation within GGAN and GAAE. To embark on this path, the initial step involves improving CNN models for the modelling of raw audio, particularly in tasks like audio classification.

In this thesis, I further explore this avenue and introduce two novel CNN-based models designed for the direct classification of audio from raw waveforms. The first model, known as the Cosine Convolutional Neural Network (CosCovNN), distinguishes itself by replacing the conventional CNN filter with a cosine-based filter, achieving superior performance with approximately 77% fewer parameters compared to similar CNN models. Furthermore, I present an enhanced version of CosCovNN, named Vector Quantised Cosine Convolutional Neural Network with Memory (VQCCM), which incorporates vector quantisation and memory modules. My extensive investigations into VQCCM's performance on various audio classification tasks, using diverse audio datasets, have resulted in setting new benchmarks for most of the studies. These models collectively open up exciting avenues for future research, exploring their potential to generate raw audio within the GGAN and GAAE framework.

1.2. Research Aim and Objectives

The primary aim of this research is to develop new machine learning models that can learn distinct and meaningful patterns (disentangled representations) from large amounts of audio data that do not have labels. This is particularly useful for improving tasks that follow, like classifying emotions in audio clips. The models are specifically designed to work with visual representations of audio, such as spectrograms.

In addition to the primary aim, this research also has a secondary aim: to create models capable of directly processing raw audio waveforms, reducing the dependency on spectrogram representations. To achieve these aims, three specific objectives have been identified:

1. Create a model that can learn useful patterns from unlabelled audio data with the help of a small amount of labelled data. This approach focuses on learning patterns that are specific to a particular task, such as recognising emotions in audio.

2. Develop a model that not only learns patterns specific to one task but also learns patterns that can be useful for other tasks. This helps in making the model more versatile and applicable to different types of problems, even those it was not originally trained for.
3. Create a model that can work directly with raw audio data instead of relying on visual representations like spectrograms. This reduces the need for complex data processing and allows the model to work with audio in its original form.

The first two objectives align with the primary aim of disentangled representation learning by acquiring task-specific and transferable representations from unlabelled audio data. Furthermore, the third objective aligns with the secondary aim of reducing dependence on spectrogram representations by enabling direct processing of raw audio waveforms.

1.3. Contributions and Outline

This research significantly contributes to machine learning and audio processing by addressing the challenge of extracting clear and useful patterns from large sets of unlabelled audio data. It also explores new methods for working directly with raw audio, avoiding the need for complex data transformations. The main contributions and outline of this thesis are as follows:

Chapter 3: I present a new model that uses a technique called Generative Adversarial Networks (GANs) to learn specific patterns from unlabelled audio data. This model helps improve the performance of related tasks, such as classifying emotions in audio, by using a small amount of labelled data to guide the learning process.

Chapter 4: I introduce a model that combines learning for both specific and general purposes. It learns detailed patterns useful for a particular task, as well as broader patterns that can be applied to different tasks, all by using a small amount of labelled data to guide the process.

Chapter 5: I propose a new approach for classifying raw audio data that improves on traditional methods. This model uses special filters that perform better than regular CNN models and require fewer resources. I also introduce an advanced version that includes additional features to further enhance its performance.

These models collectively contribute to the fields of machine learning and audio processing by providing versatile tools that enhance the ability to leverage unlabelled data, streamline model development, and address complex challenges effectively. The implications of this research extend beyond audio processing, offering opportunities to simplify and advance artificial intelligence across various domains.

1.4. Outcomes and Implications

The results of this research hold significant implications for the fields of machine learning and audio processing. One key outcome is the ability to leverage vast amounts of unlabelled data effectively. This empowers machine learning practitioners to enhance models, even when labelled data is scarce. The models, initially designed for audio, are adaptable and can be applied to 2D audio representations, expanding their utility to domains like computer vision.

Another crucial outcome is the development of models designed to process raw audio waveforms directly. This contrasts with traditional methods, which rely heavily on manual feature engineering and complex data preparation. The models reduce computational demands and simplify the research process, accelerating scientific progress.

In summary, this research represents a significant step forward in deep representation learning. The models provide versatile tools for researchers across various fields, enhancing their ability to address complex challenges effectively and push the boundaries of artificial intelligence. The implications of this research extend far beyond audio processing, offering opportunities to leverage unlabelled data, streamline model development, and advance artificial intelligence across diverse domains.

CHAPTER 2: LITERATURE REVIEW

2.1. Introduction

This chapter is dedicated to discussing the relevant literature to provide background knowledge and set the stage for this research. Subsequent chapters will each conduct their own literature review, focusing on their specific domain.

2.2. Deep Learning

Deep learning has significantly impacted the field of machine learning in recent years, primarily due to advancements in computational power that enable the training of complex neural networks for a variety of tasks [24]. The interest in deep learning - surged among researchers particularly after 2012, when a team led by Geoffrey Hinton achieved a breakthrough by winning the ImageNet competition, showcasing the potential of Convolutional Neural Networks (CNNs) in computer vision tasks [25]. This victory underscored the efficiency of deep learning models over traditional machine learning approaches in analysing visual data.

The success of deep learning extends beyond computer vision; it has revolutionised several other domains. In machine translation, deep learning algorithms have enhanced the quality and efficiency of translating text between languages, achieving remarkable fluency and accuracy [26-28]. Speech recognition and speech-to-text conversion have also seen substantial improvements, making interactions with voice-activated systems more seamless and natural [29-31]. Deep learning has enabled the creation of models in natural language processing that generate realistic text [32, 33] and descriptive image captions [34-36], merging visual content understanding with language comprehension. Furthermore, deep learning has made strides in areas like video understanding [37-39], enabling more sophisticated analysis of video content for applications such as content categorisation and activity recognition. Recently, Vision Transformers (ViTs) have emerged as a powerful alternative to CNNs, achieving superior performance on various computer vision tasks and showing potential in audio representation learning. The hierarchical structure of models like the Swin Transformer enables efficient processing of complex data, paving the way for further exploration in domains beyond vision [40-43].

It has also been applied to generate and synthesize images and videos [12, 16, 44-46], creating realistic and high-quality outputs that can be used in various applications, from entertainment to educational content creation. The application of deep learning has also extended to more specialized fields such as medical diagnostics [47-49], where it assists in identifying patterns in imaging data that are indicative of specific diseases, and biotechnology, where it aids in analysing complex biological data. In addition, deep learning has been utilised in creative domains, enabling the generation of art [50], music [51-53], and even writing code [54, 55], showcasing its versatility and adaptability across different creative processes.

Given the broad application of deep learning across various sectors, it has become a cornerstone of modern AI research and development. Its ability to outperform traditional models in a wide range of tasks has made it an indispensable tool in pushing the boundaries of what machines can learn and accomplish. As computational resources continue to evolve and become more accessible, the potential for deep learning to drive innovation and solve complex problems grows exponentially, making it a dynamic and continually evolving field of study.

2.3. Supervised Transfer Learning

The advent of deep neural networks has ushered in significant advancements in supervised learning. Among these, supervised transfer learning has garnered attention for its efficacy and versatility [7]. At its core, transfer learning involves the process of pretraining a neural network on a source task before fine-tuning it on a target task, which may involve either classification or regression. This methodology is particularly beneficial as the weights obtained during pretraining facilitate generalisation, allowing for the fine-tuning phase to effectively utilise limited label information to adjust the weights for the target task. Transfer learning is predicated on the notion that, although the training datasets for the source and target tasks may differ in their statistics, the representation learned from the source dataset can enhance performance on the target task [56]. This is evident in applications such as object detection [57], scene classification [58], semantic segmentation [59], image captioning [60], and Audio Classification [31, 61-63], where networks pretrained on large datasets exhibit remarkable adaptability to new tasks. Yosinski et al. highlighted the potential of transferred features from pretrained networks to improve generalisation and performance on subsequent tasks [64].

The concept of transfer learning extends beyond visual tasks, finding applications in natural language processing (NLP) as well. For instance, cross-lingual document classification benefits from transfer learning by utilizing classifiers trained on data in one language to perform tasks in another [65]. Gouws et al. demonstrated that effective word representations can be learned from monolingual text and a limited amount of parallel data, setting new benchmarks in English-German cross-lingual classification without the need for word alignments or dictionaries [66].

Transfer learning also encompasses more challenging paradigms such as one-shot and zero-shot learning. In one-shot learning, the model is trained with a single example per class, whereas zero-shot learning involves classes that are not present in the training data at all. These approaches highlight the potential of transfer learning to generalise from minimal data, offering solutions for tasks with scarce labelled data [7, 67, 68].

While supervised transfer learning has proven effective in leveraging labelled data across different tasks, its reliance on labelled data constitutes a limitation. Specifically, it lacks the capability to utilise unlabelled data, which is abundantly available and potentially useful for learning. This gap underscores the necessity for exploring alternative approaches that can incorporate both labelled and unlabelled data to enhance learning efficacy and model performance. Recent advances in contrastive learning have significantly enhanced transfer learning by enabling models to learn more robust and generalizable representations from unlabelled data. Approaches like CLIP leverage natural language supervision to achieve cross-domain adaptability, including applications in audio [69-72].

2.4. Unsupervised Representation Learning

In addressing the challenges posed by the necessity for large volumes of labelled data in supervised transfer learning, the focus has shifted towards Unsupervised Representation Learning [8, 73, 74]. This method is distinguished by its ability to utilise unlabelled data to learn comprehensive representations of the underlying data distribution. Such representations prove extremely useful when applied to tasks where labelled data is scarce, enabling models to perform effectively with minimal supervision [75].

Unsupervised Representation Learning is defined by its methodology of extracting features and patterns from data without any labels guiding the process. This

approach allows the model to capture the intrinsic properties of the dataset, facilitating a deeper understanding of its structure. These learned representations are then applicable to a variety of tasks, demonstrating the versatility of unsupervised learning in leveraging unlabelled data [4].

A prime example of this methodology's success is seen in the domain of Large Language Models (LLMs) [76, 77]. These models undergo a process of unsupervised pretraining on extensive text corpora, during which they learn rich linguistic representations. The beauty of this approach lies in its next step: fine-tuning the pre-trained models on smaller, task-specific labelled datasets. This two-phase process showcases the efficiency of unsupervised representation learning in making the most of the abundant unlabelled data available, subsequently applying this knowledge to enhance performance on tasks with limited labelled examples.

This paradigm shifts towards unsupervised pretraining followed by task-specific fine-tuning has not only mitigated the reliance on large, labelled datasets but has also broadened the scope of machine learning applications. By effectively utilising unlabelled data, unsupervised representation learning paves the way for models to achieve high levels of performance across a wide array of tasks, even when faced with the challenge of limited supervision. The rise of self-supervised learning has further bridged the gap between supervised and unsupervised learning, particularly in the realm of audio and language models. Methods like Bootstrap Your Own Latent (BYOL) and Data2vec demonstrate the power of self-supervised learning in extracting useful representations from unlabelled data [78-81].

2.4.1. Traditional Methods

Unsupervised representation learning, a key area of focus within machine learning, has its roots deeply embedded in the field's history. The theoretical benefits of this approach were first articulated by Hinton in 1986, marking a pivotal moment in understanding the potential of learning representations without direct supervision [82]. This foundational concept found practical application through the training of neural networks, where early successes underscored the viability and importance of unsupervised pretraining [7]. The strategy of Greedy layer-wise pretraining, in particular, proved instrumental in the development of deep belief networks [83] and in optimizing deep autoencoder networks to learn compact, low-dimensional representations [84].

The significance of pretraining extends beyond the mere initialisation of network weights; it introduces robustness to deep neural networks, protecting against the entrapment in suboptimal local minima—a frequent obstacle in neural network training [85]. Furthermore, the qualitative differences in features produced by pretrained networks as opposed to those without pretraining highlight the transformative impact of this approach on the network's ability to capture and represent complex data structures [86]. This methodology has also been successfully applied to Deep Boltzmann Machines, where sequential layer-by-layer pretraining enhances the efficiency of variational inference [87] demonstrating the effectiveness of unsupervised pretraining in leveraging the wealth of unlabelled data.

The exploration of unsupervised representation learning is not limited to neural networks but extends to classical machine learning models. Principal Component Analysis (PCA), proposed by Pearson in 1901, stands as a foundational method for learning low-dimensional representations of data [88] , its simplicity making it a go-to technique for dimensionality reduction [89]. While Linear Discriminant Analysis (LDA) was initially introduced as a supervised method [90], the field has seen the emergence of nonlinear approaches such as kernel PCA [91] and Generalised Discriminant Analysis (GDA) [92], each contributing to the nuanced understanding of data representation.

Further advancements in representation learning techniques have included the development of Marginal Fisher Analysis (MFA) by Yan et al., which emphasizes the distinction between intraclass compactness and interclass separability through graph embeddings, positioning MFA as a versatile algorithm for discriminant analysis [93]. Comparative analyses have highlighted the superior recognition accuracy of MFA amidst its higher complexity, whereas methods like LDA, despite their efficiency, may not always yield optimal results [94].

The field has also benefited from clustering approaches, as demonstrated by Coates et al., who achieved state-of-the-art results by employing K-means clustering alongside pre-identified network parameters for efficient representation learning [95] , and further advancements in hierarchical clustering for image patches [96]. However, the scalability of these traditional methods remains a challenge, particularly as they are generally more applicable to smaller datasets, underscoring a critical area for further research and development in unsupervised representation learning [97].

This journey from the early theoretical propositions to the diverse array of techniques available today illustrates the continuous evolution and growing complexity of unsupervised representation learning. The field stands on the brink of further discoveries, with each method contributing to the overarching goal of harnessing the vast potential of unlabelled data in learning meaningful and efficient representations.

2.4.2. Autoencoders and Generative Adversarial Neural Networks

Autoencoders have played a pivotal role in advancing unsupervised representation learning, with Hinton demonstrating that through the use of log-linear activation functions and a reduction in code size, autoencoders could learn valuable features in an unsupervised manner [84]. Denoising autoencoders, which are trained to remove noise from corrupted inputs, further improved the ability to capture higher-level representations, aiding in various supervised tasks such as sentiment analysis [98-100]. The introduction of stacked denoising autoencoders and the integration of convolutional networks as encoders and deconvolutional networks as decoders marked significant improvements in representation learning [101, 102]. Variants like the Contractive Autoencoder and Split-Brain Autoencoder have also demonstrated their utility in initializing deep architectures and achieving state-of-the-art performance in transfer learning benchmarks, respectively [103, 104].

The advent of Variational Autoencoders (VAEs) brought a generative aspect to autoencoders, with researchers focusing on VAEs for their ability to model posterior distributions and conditional log-likelihoods in a probabilistic framework [105]. Improvements to VAEs aimed at learning more expressive posterior distributions have been proposed, addressing limitations such as the tendency of VAEs to ignore the latent code [106, 107]. Furthermore, the introduction of Adversarial Autoencoders and PixelGANs, which incorporate adversarial training, has led to breakthroughs in semi-supervised classification, unsupervised clustering, and more [108-110].

Generative Adversarial Networks (GANs) have emerged as a cornerstone in the field of unsupervised representation learning due to their success in generating high-quality, diverse images and their potential in learning disentangled representations[5, 7, 16]. Radford et al. highlighted GANs as a promising candidate for unsupervised learning, capable of vector arithmetic in latent spaces previously only seen in natural language processing [5]. Recent innovations have addressed GANs' initial limitations in generating large images and capturing entire data distributions,

with Nvidia's progressive training methodology allowing for the creation of realistic images [14, 15]. Despite these advancements, capturing the full data distribution remains a challenge, with recent work by Yoshua Bengio's team proposing methods to better capture this distribution [111, 112].

GANs have also begun to make inroads in the audio domain, overcoming challenges in generating raw audio to achieve significant successes in voice cloning tasks [113]. This opens up new avenues for research into GAN applications beyond computer vision, including audio representation learning, where there is substantial potential for innovation [114, 115]. Recently, diffusion models have emerged as a promising alternative to GANs, achieving state-of-the-art results in generative tasks, including audio synthesis. These models offer a robust approach to overcoming some of the limitations associated with GANs, such as mode collapse and training instability [116-118]. Inspired by breakthroughs in voice cloning and representation learning from raw audio, our work contributes to this growing field by developing a model that effectively learns representations from audio data through GANs, building on the successes of pioneers in the field [119].

2.5. Links and Implications

This exploration of unsupervised representation learning reveals a promising avenue for leveraging the vast reserves of unlabelled data to enhance performance in downstream tasks, particularly those constrained by limited labelled data. This methodology stands as a testament to the ingenuity of utilising inherent data structures to pre-train models, thereby circumventing the traditional reliance on extensive annotated datasets. Such an approach not only democratises the accessibility of advanced machine learning techniques across varied domains but also amplifies the potential for discoveries in fields where labelled data is scarce or expensive to procure.

However, the journey through unsupervised representation learning unveils a significant caveat—the bifurcated process of initially training on unlabelled data followed by fine-tuning on a smaller, labelled dataset. This two-step procedure introduces a layer of complexity, as the success of pre-training does not inherently guarantee efficacy in downstream applications. The performance of the model in the pre-training phase is not always indicative of its adaptability or effectiveness in subsequent tasks, presenting a challenge in predictive evaluation and optimization.

The critical evaluation of these methodologies underscores a pivotal concern: without prior insight into the specific requirements of the downstream task, there is a risk that the model may not learn the necessary features. This disconnect posits a barrier to the universal applicability of unsupervised representation learning, emphasising the need for targeted innovations that bridge this gap.

In response to these challenges, this thesis proposes novel models that harness the capabilities of generative adversarial neural networks and adversarial autoencoders. These models are designed to mitigate the limitations associated with the two-step training process, aiming to enhance the model's ability to learn relevant features with some guidance from the downstream task. By integrating adversarial mechanisms, these models strive to generate more robust and versatile representations, potentially increasing the efficacy and reliability of unsupervised learning in diverse applications.

The implications of these advancements extend far beyond the technical realm, offering a glimpse into the future of machine learning where data's intrinsic value is fully harnessed. By refining and expanding upon these unsupervised learning models, there is an opportunity to significantly reduce the barrier to entry for sophisticated data analysis, opening new pathways for innovation across scientific research, technology development, and beyond. Furthermore, the adoption of these models could catalyse a shift towards more efficient and adaptable machine learning frameworks, promising to reshape the landscape of data-driven inquiry and application.

As we advance, the continual refinement and validation of these proposed models will be paramount. The journey through unsupervised representation learning, with its trials and triumphs, not only enriches our understanding of the potential within unlabelled data but also sets the stage for future explorations that may one day render the scarcity of labelled data a negligible concern in the pursuit of knowledge and innovation.

CHAPTER 3: PAPER 1 – Guided Generative Adversarial Neural Network for Representation Learning and Audio Generation using Fewer Labelled Audio Data

3.1. Introduction

This chapter primarily addresses objective 1 and introduces a novel GAN-based model called the Guided Generative Adversarial Neural Network (GGAN). GGAN is designed to generate high-quality conditional audio samples using unlabelled audio datasets, guided by a limited number of labelled datasets. Throughout the conditional generation process, the model learns to disentangle the latent space based on the classification signal provided by the accompanying labelled dataset, thereby enhancing its performance on this labelled dataset. I evaluated the model based on both speech and nonspeech datasets and proved that using only 5% of labelled data as guidance, GGAN learns significantly better representations than the state-of-the-art models.

3.2. Published paper

3.3. Links and implications

This chapter demonstrated that GGAN can achieve good accuracy in audio tasks when limited labelled data is available, thanks to its utilisation of a substantial amount of related unlabelled audio data. GGAN leverages unlabelled data to acquire task-specific representations, but these representations are primarily suited for specific downstream task**s** or closely related tasks. Despite its promising performance, GGAN faces limitations in scaling to unfamiliar tasks that are not related to the labelled data used during training. To address this issue, the following chapter presents an alternative model capable of simultaneously learning both task-specific and generalised representations, thus bridging this gap.

CHAPTER 4: PAPER 2 - High-Fidelity Audio Generation and Representation Learning with Guided Adversarial Autoencoder

4.1. Introduction

In this chapter, I delve into the second objective of the research and present the innovative Guided Adversarial Autoencoder (GAAE) model. GAAE is structured around an encoder and decoder architecture. The decoder's role is to learn and generate high-quality audio samples that effectively capture the diverse mode within the training data distribution. It achieves this by incorporating guidance from a limited subset of labelled data, either from the same dataset or a closely related one.

The key strength of GAAE lies in its ability to produce high-fidelity audio samples, empowering the encoder to disentangle specific data attributes within the learned latent or representation space according to the provided guidance. This acquired representation proves valuable for enhancing any related downstream task at hand. Additionally, I demonstrate that GAAE extends beyond guided representation learning, uncovering, and disentangling additional data attributes that remain independent of the provided guidance. As a result, GAAE excels in simultaneously learning task-specific representations tailored to the immediate downstream task, while also acquiring generalised representations capable of accommodating unforeseen, unrelated tasks in the future.

4.2. Published paper

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 4, 2020, accepted November 14, 2020, date of publication November 26, 2020,
date of current version December 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3040797

High-Fidelity Audio Generation and Representation Learning With Guided Adversarial Autoencoder

KAZI NAZMUL HAQUE[1], RAJIB RANA[1], (Member, IEEE), AND BJÖRN W. SCHULLER, JR.[2,3], (Fellow, IEEE)

[1]Department of Computer Science, School of Science, University of Southern Queensland, Toowoomba, QLD 4301, Australia
[2]Group on Language, Audio and Music (GLAM), Imperial College London, London SW7 2AZ, U.K.
[3]Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg, 86159 Augsburg, Germany

Corresponding author: Kazi Nazmul Haque (shezan.huq@gmail.com)

ABSTRACT Generating high-fidelity conditional audio samples and learning representation from unlabelled audio data are two challenging problems in machine learning research. Recent advances in the Generative Adversarial Neural Networks (GAN) architectures show great promise in addressing these challenges. To learn powerful representation using GAN architecture, it requires superior sample generation quality, which requires an enormous amount of labelled data. In this paper, we address this issue by proposing Guided Adversarial Autoencoder (GAAE), which can generate superior conditional audio samples from unlabelled audio data using a small percentage of labelled data as guidance. Representation learned from unlabelled data without any supervision does not guarantee its' usability for any downstream task. On the other hand, during the representation learning, if the model is highly biased towards the downstream task, it losses its generalisation capability. This makes the learned representation hardly useful for any other tasks that are not related to that downstream task. The proposed GAAE model also address these issues. Using this superior conditional generation, GAAE can learn representation specific to the downstream task. Furthermore, GAAE learns another type of representation capturing the general attributes of the data, which is independent of the downstream task at hand. Experimental results involving the S09 and the NSynth dataset attest the superior performance of GAAE compared to the state-of-the-art alternatives.

INDEX TERMS Audio generation, representation learning, generative adversarial neural network, guided generative adversarial autoencoder.

I. INTRODUCTION

Representation learning aims to map higher-dimensional data into a lower-dimensional representation space where the variational factors of the data are disentangled. Learning a disentangled representation from an unlabelled dataset opens a window of opportunity for researchers to utilise the vastly available unlabelled data for any downstream tasks [1]. Such as, a representation learnt from freely available YouTube audios (movie, news etc.) can be used to improve a task such as emotion recognition from audio where a large labelled dataset is unavailable.

Generative Adversarial Neural Network (GAN) [2] has shown great promise for learning powerful representation. GAN is comprised of a Generator network and a Discriminator network, where these networks are trained to defeat each other based on a minimax game. During training, the Generator tries to fool the Discriminator by generating real-like samples from a random noise/latent distribution, and the Discriminator tries to defeat the Generator by differentiating the generated sample from the real samples [2]. During this game-play, the Generator disentangles the underlying attributes of the data in the given random latent distribution [3]. This helps in learning powerful representations [3]–[9] in a unsupervised manner. GAN based models pose great promise in audio research where limited or no labelled data is available.

The representation learning performance of the GANs usually improves along with its' sample generation quality. Intuitively, GAN models that can generate high-quality

The associate editor coordinating the review of this manuscript and approving it for publication was Ananya Sen Gupta.

samples, intrinsically learns powerful representation [6]. GAN-based models are successful at generating high-fidelity images, however, they fail to perform likewise for the complex audio waveform generation as it requires modelling higher-order temporal scales [10]. To successfully generate audio with GANs, many researchers have worked with the spectrogram of the audio which can be converted back to the audio with minimal loss [10]–[12]. Recently proposed high performing GAN architectures such as BigGAN [13] and StyleGAN [9] are not well explored in the audio field, leaving a room to explore the compatibility of these models for audio data.

A representation learnt with GANs in a completely unsupervised manner does not guarantee the usability of the learnt representation for any particular downstream task. This is because it can ignore the important characteristics of the data during the training which is important for succeeding in the downstream task [14]. So, some bias towards the downstream task is necessary during the unsupervised training to succeed in that downstream task [1].

GAN models perform better for conditional generation using labelled data. The labels add useful side information during the training, which helps the GAN models to decompose overall sample generation tasks into sub-tasks according to the conditioned labels. Though the conditional generation helps to improve performance significantly, it requires an enormous amount of labelled data [15], which is costly and/or error-prone. Using the GAN models to generate high-quality samples with a minimum amount of labelled data therefore remains a crucial challenge [14].

In our previous work, we propose a BigGAN based architecture called "Guided Generative Adversarial Neural Network (GGAN)," which can generate state-of-the-art (SOTA) conditional audio with fewer labelled data. This labelled data is used as a guidance to force GGAN to learn guided representation for any downstream task at hand. Note that, the learned representation for any particular downstream task makes it less useful for any other task that is unrelated to the downstream task [14]. In many cases, it is desirable to learn representation in a manner so that it can be used for any particular downstream task as well as can be used for any future tasks independent of the downstream task at hand [16]. It is a challenging problem to learn both generalised and guided representation at the same time with conditional GAN architectures. During the training of any conditional GAN, the latent noise/samples are independent of the given condition. So, GAN learns to map the general characteristics of the training data from the latent samples, which is independent of the condition. On the other hand, if the condition is imposed on the latent samples/noise like GGAN, the latent cannot learn general characteristics as it is biased towards the conditioned attributes. In this paper, we address this problem. Our contributions are as follows:

- We propose a novel autoencoder based GAN model GAAE, which can generate high-fidelity audio samples capturing the diverse modes of the training data distribution leveraging the guidance from a fewer labelled data samples from that dataset or a related dataset.
- We evaluate the conditional sample generation quality of the proposed model based on two audio datasets: the Speech Command dataset (S09) and the Musical Instrument Sound dataset (Nsyth). We demonstrate that the GAAE model performs significantly better than the SOTA models.
- We achieve generalised and guided representation in our GAAE model. Evaluation results on three different datasets: the Speech Command dataset (S09), the Audio Book Speech dataset (Librispeech), and the Musical Instrument Sound dataset (Nsyth) show that the proposed GAAE model performs better than SOTA models.

II. BACKGROUND AND RELATED WORK

A. AUDIO REPRESENTATION LEARNING

While there is a rich literature of supervised representation learning, due to our focus on unsupervised representation learning we will only discuss the related literature here. In the field of unsupervised representation learning, the self-supervised learning has become very popular recently due to its unprecedented success in the field of computer vision [17]–[23] and natural language processing [24]–[27]. Self-supervised learning uses information presents in the unlabelled data to create an alternative supervised signal to train the model for learning feature/representation. For an example, learning representation through predicting the rotation angel of images where rotation angel serves as supervised signal and this learned representation can be used to improve other related image classification tasks [28].

Likewise, in the audio field, researchers have achieved good performances using self-supervised representation learning. In their work, DeepMind [29] have proposed a model to learn a useful representation from unsupervised speech data through predicting a future observation in the latent space. In another work from Google [30], the representation is learnt by predicting the instantaneous frequency based on the magnitude of the Fourier transform. Furthermore, Arsha *et al.* (2020) [31] proposed a cross-modal self-supervised learning method to learn speech representation from the co-relationship between the face and the audio in the video. Other efforts have been made by researchers to learn a general representation by predicting the contextual frames of any particular audio frame like wav2vec [32], speech2vec [33], and audio word2vec [34]. Likewise, there are other successful implementations [35]–[38] of the self-supervised representation learning in the field of audio.

Though self-supervised learning is good for learning representations from unlabelled datasets, it requires manual endeavour to design the supervision signal [39]. To avoid this, researchers have focused on fully unsupervised representation learning mainly using autoencoders [40]–[42]. In [43], the authors learnt representations with an autoencoder

from a large unlabelled dataset, which improved the emotion recognition from speech audio. Similarly, in another work, the authors used a denoising autoencoder to improve affect recognition from speech data [44]. Several works [5], [45], [46] have utilised Variational Autoencoders (VAEs) [47] to learn an efficient speech representation from an unlabelled dataset. Recently, given the popularity of adversarial training, different works have been conducted by researchers to learn a robust representation with GANs [48], [49] and Adversarial Autoencoders [50], [51].

Though learning a representation from prodigiously available unlabelled datasets is very intriguing, the recent work from Google AI has proved that completely unsupervised representation learning is not possible without any form of supervision [1]. Also, representation learnt from an unsupervised method does not guarantee the usability of this learnt representation for any post use case scenario. Thus, as outlined, we proposed the Guided Generative Adversarial Neural Network (GGAN) [14], which can learn a powerful representation from an unlabelled audio dataset according to the supervision given from a fewer amount of labelled data. Therefore, in the learnt representation space, the GGAN disentangles attributes of the data according to the given categories from the labelled dataset, which benefits the related post-use case scenario.

B. AUDIO GENERATION

Most of the audios are periodic, and high-fidelity audio generation requires modelling a higher order magnitude of the temporal scales, which makes it a challenging problem [10]. Most of the research works related to audio generation are based on the audio synthesis viz; Aaron and *et al.* (2016) have proposed a powerful autoregressive model named "Wavenet," which works very well on text to speech (TTS) synthesis for both English and Mandarin. Later, the authors have improved this work by proposing "Parallel Wavenet," which is 20 times faster than the original Wavenet. Other researchers have utilised the seq2seq model for TTS such as Char2Wav [52] and TACOTRON [53]. However, these audio generation methods are conditioned on the text data and mainly focused on speech generation. Thus, these methods cannot be generalised to all other audio domains, even for speech data where transcripts are not available.

In the context of generating audio without any condition on the text data, the GANs are very promising due to their massive success in the field of computer vision [6], [9], [54]–[56]. However, porting these GAN architectures directly to the audio domain does not offer similar performance as the audio waveform is mostly more complex than an image [10], [11]. Therefore, researchers have focused on generating spectrogram (2D image-like representation of audio) rather than generating directly a waveform. Then, the generated spectrogram is converted back to audio. Chris *et al.* (2019) [11] have trained a GAN-based model to generate spectrograms and successfully converted them back to the audio domain with the Griffin-Lim algorithm [57]. In their TiFGAN paper [12], the authors have proposed a phase-gradient heap integration (PGHI) [58] algorithm for better reconstruction of the audio from the spectrogram with minimal loss. As the PGHI algorithm is good at reconstructing audio from the spectrogram, now the challenge is to generate a realistic spectrogram. As the spectrogram is—as outlined—an image-like representation of the audio, any GAN based framework from the image domain should be compatible. Hence, the BigGAN architecture [13] has shown promising performance at generating conditional high resolution/fidelity images, but it was not well explored for audio generation. In this paper we address this gap.

C. CLOSELY RELATED ARCHITECTURES

The proposed GAAE model is a semi-supervised model, as we leverage a small amount of labelled data during the training. In [59], the authors proposed a semi-supervised version of the InfoGAN model [4] to capture a specific representation and generation according to the supervision which comes from a small number of labelled data. But, the success of this model in terms of the complex data distribution is not evident. Other researchers have explored the scope of semi-supervision in GAN architectures [15], [60], [61] to improve the conditional generation, but most of these works are not explored in the audio domain which leaves a major gap for the researchers to address. The GAAE model is based on an Adversarial Autoencoder (AAE) [8], where we have extended the AAE model to learn both guided and generalise/style representation from an unlabelled dataset in a semi-supervised fashion. Furthermore, in the GAAE model, we have implemented a unique way to leverage the small amount of labelled data for conditional audio generation. Here, we have also proposed a way to utilise the generated conditional samples for improving the representation learning during the training. Moreover, the building block for our GAAE model is a BigGAN architecture; thus, we further contribute by exploring the use of a BigGAN in an autoencoder-based model for audio data.

D. AUDIO REPRESENTATION LEARNING

While there is a rich literature of supervised representation learning, due to our focus on unsupervised representation learning we will only discuss the related literature here. In the field of unsupervised representation learning, the self-supervised learning has become very popular recently due to its unprecedented success in the field of computer vision [17]–[23] and natural language processing [24]–[27]. Self-supervised learning uses information presents in the unlabelled data to create an alternative supervised signal to train the model for learning feature/representation. For an example, learning representation through predicting the rotation angel of images where rotation angel serves as supervised signal and this learned representation can be used to improve other related image classification tasks [28].

Likewise, in the audio field, researchers have achieved good performances using self-supervised representation

learning. In their work, DeepMind [29] have proposed a model to learn a useful representation from unsupervised speech data through predicting a future observation in the latent space. In another work from Google [30], the representation is learnt by predicting the instantaneous frequency based on the magnitude of the Fourier transform. Furthermore, Arsha *et al.* (2020) [31] proposed a cross-modal self-supervised learning method to learn speech representation from the co-relationship between the face and the audio in the video. Other efforts have been made by researchers to learn a general representation by predicting the contextual frames of any particular audio frame like wav2vec [32], speech2vec [33], and audio word2vec [34]. Likewise, there are other successful implementations [35]–[38] of the self-supervised representation learning in the field of audio.

Though self-supervised learning is good for learning representations from unlabelled datasets, it requires manual endeavour to design the supervision signal [39]. To avoid this, researchers have focused on fully unsupervised representation learning mainly using autoencoders [40]–[42]. In [43], the authors learnt representations with an autoencoder from a large unlabelled dataset, which improved the emotion recognition from speech audio. Similarly, in another work, the authors used a denoising autoencoder to improve affect recognition from speech data [44]. Several works [5], [45], [46] have utilised Variational Autoencoders (VAEs) [47] to learn an efficient speech representation from an unlabelled dataset. Recently, given the popularity of adversarial training, different works have been conducted by researchers to learn a robust representation with GANs [48], [49] and Adversarial Autoencoders [50], [51].

Though learning a representation from prodigiously available unlabelled datasets is very intriguing, the recent work from Google AI has proved that completely unsupervised representation learning is not possible without any form of supervision [1]. Also, representation learnt from an unsupervised method does not guarantee the usability of this learnt representation for any post use case scenario. Thus, as outlined, we proposed the Guided Generative Adversarial Neural Network (GGAN) [14], which can learn a powerful representation from an unlabelled audio dataset according to the supervision given from a fewer amount of labelled data. Therefore, in the learnt representation space, the GGAN disentangles attributes of the data according to the given categories from the labelled dataset, which benefits the related post-use case scenario.

E. AUDIO GENERATION

Most of the audios are periodic, and high-fidelity audio generation requires modelling a higher order magnitude of the temporal scales, which makes it a challenging problem [10]. Most of the research works related to audio generation are based on the audio synthesis viz; Aaron and *et al.* (2016) have proposed a powerful autoregressive model named "Wavenet," which works very well on text to speech (TTS) synthesis for both English and Mandarin. Later, the authors have improved this work by proposing "Parallel Wavenet," which is 20 times faster than the original Wavenet. Other researchers have utilised the seq2seq model for TTS such as Char2Wav [52] and TACOTRON [53]. However, these audio generation methods are conditioned on the text data and mainly focused on speech generation. Thus, these methods cannot be generalised to all other audio domains, even for speech data where transcripts are not available.

In the context of generating audio without any condition on the text data, the GANs are very promising due to their massive success in the field of computer vision [6], [9], [54]–[56]. However, porting these GAN architectures directly to the audio domain does not offer similar performance as the audio waveform is mostly more complex than an image [10], [11]. Therefore, researchers have focused on generating spectrogram (2D image-like representation of audio) rather than generating directly a waveform. Then, the generated spectrogram is converted back to audio. Chris *et al.* (2019) [11] have trained a GAN-based model to generate spectrograms and successfully converted them back to the audio domain with the Griffin-Lim algorithm [57]. In their TiFGAN paper [12], the authors have proposed a phase-gradient heap integration (PGHI) [58] algorithm for better reconstruction of the audio from the spectrogram with minimal loss. As the PGHI algorithm is good at reconstructing audio from the spectrogram, now the challenge is to generate a realistic spectrogram. As the spectrogram is—as outlined—an image-like representation of the audio, any GAN based framework from the image domain should be compatible. Hence, the BigGAN architecture [13] has shown promising performance at generating conditional high resolution/fidelity images, but it was not well explored for audio generation. In this paper we address this gap.

F. CLOSELY RELATED ARCHITECTURES

The proposed GAAE model is a semi-supervised model, as we leverage a small amount of labelled data during the training. In [59], the authors proposed a semi-supervised version of the InfoGAN model [4] to capture a specific representation and generation according to the supervision which comes from a small number of labelled data. But, the success of this model in terms of the complex data distribution is not evident. Other researchers have explored the scope of semi-supervision in GAN architectures [15], [60], [61] to improve the conditional generation, but most of these works are not explored in the audio domain which leaves a major gap for the researchers to address. The GAAE model is based on an Adversarial Autoencoder (AAE) [8], where we have extended the AAE model to learn both guided and generalise/style representation from an unlabelled dataset in a semi-supervised fashion. Furthermore, in the GAAE model, we have implemented a unique way to leverage the small amount of labelled data for conditional audio generation. Here, we have also proposed a way to utilise the generated conditional samples for improving

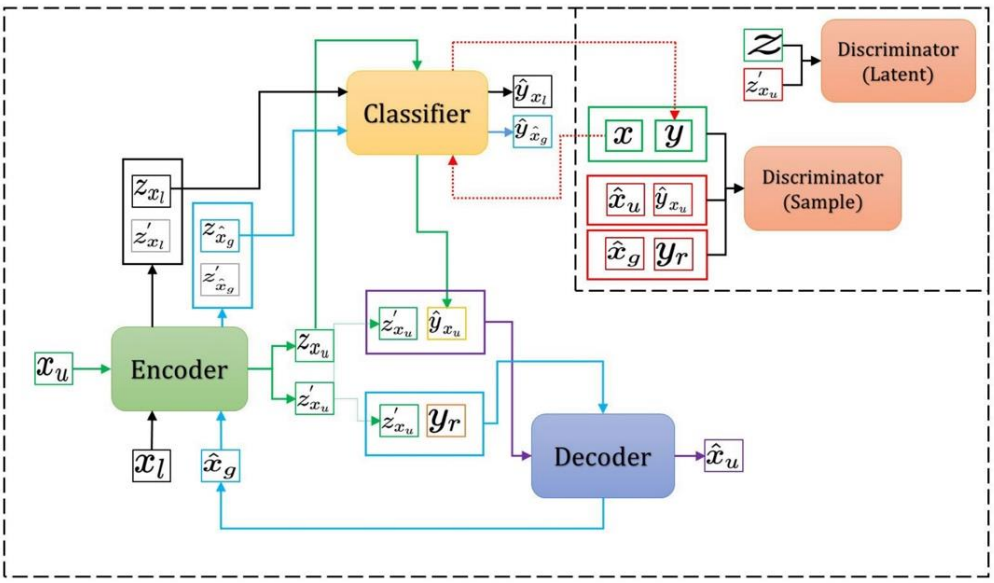

FIGURE 1. This figure illustrates the overall architecture of the GAAE model. Different networks of the GAAE model are shown along with the connections between them. In the figure, the arrows are coloured to highlight the flow of any input/output of the model. For the discriminator, the red boxes show the fake samples and the green boxes indicate the real samples. Here, x_u is the unlabelled data sample, x_l is the labelled data sample,$\hat{x}_u$ is the reconstructed data sample, y_r is the random conditions, and z is the known latent distribution.

the representation learning during the training. Moreover, the building block for our GAAE model is a BigGAN architecture; thus, we further contribute by exploring the use of a BigGAN in an autoencoder-based model for audio data.

III. PROPOSED RESEARCH METHOD

A. ARCHITECTURE OF THE GAAE

The GAAE consists of five neural networks: the Encoder E, the Decoder D, the Classifier C, the Latent Discriminator L and the Sample Discriminator S. Let the parameters for these networks be $\theta_e, \theta_d, \theta_c, \theta_L$, and θ_S respectively. Figure 1 shows the whole architecture of the model and the description is as follows.

1) ENCODER

The Encoder E takes any unlabelled data sample $x_u \sim p_{data}$ and outputs two latent samples $z_{x_u} \sim u_z$ and $z'_{x_u} \sim q_z$, where p_{data} is the true unlabelled data distribution, and u_z,q_z are two different continuous distributions learned by the E. We require the latent z_{x_u} to capture the post-task-specific attributes/characteristics of the data and the latent z'_{x_u} to capture the general/style attributes of the data.

2) CLASSIFIER

We have a classifier network C which is trained with limited labelled data $x_l \sim p_{ldata}$, where p_{ldata} is the labelled data distribution and not necessarily $p_{ldata} \subset p_{data}$. Here, with this p_{ldata}, the whole model gets guidance—thus, we call this data as "guidance data." Now, the C network takes any latent sample and predicts the category class for that latent sample. To train C, we pass x_l through the E network and get two latent vectors $\{z_{x_l}, z'_{x_l}\} = E(x_l; \theta_e)$. Then, we only forward z_{x_l} through C to get the predicted label $\hat{y}_{x_l} = C(z_{x_l}; \theta_c)$ and train C against the true label $y_l \sim Cat(y_l, k = n)$ of the sample x_l, where $Cat(y_l, k = n)$ is the categorical distribution with n numbers of categories/labels. These labels are used as one-hot vector. For now, lets consider that C can classify the label of any sample correctly.

3) DECODER

The Decoder D maps any latent and categorical class/label variable to the data sample. Now, to get the reconstructed sample of x_u, we pass the latent z'_{x_u} and the label of x_u through the D network. As x_u is an unlabelled data sample, we get the label $\hat{y}_{x_u} = C(z_{x_u}, \theta_c)$ through the network C and obtain the reconstructed sample $\hat{x}_u = D(z'_{x_u}, \hat{y}_{x_u}; \theta_d)$ from the D network. Here, we also want to use the D network for generating samples according to the given condition along with the reconstruction. Therefore, the same latent z'_{x_u} is used with a random categorical variable (one-hot vector) y_r, sampled from categorical distribution $Cat(y_r, K = n, p = \frac{1}{n})$, where n is the number of categories/labels, and the sampling probability for each category is $\frac{1}{n}$. Now, we obtain the generated sample $\hat{x}_g \sim p_{gdata}$, where p_{gdata} is the generated data

distribution by the D network, and it is trained to match p_{gdata} with the true data distribution p_{data}. Here, the size of n is the same as of the guided data, and we want the D network to generate data according to the categories from the guided data. Therefore, we ensure this with the Discriminator where the Discriminator receives the labels of the data from the network C. As we use a small number of labelled data, it is hard to train C due to the problem of overfitting. Hence, we use the generated sample $\hat{x}_g$ and train the C network considering y_r as the true label/category, where the predicted label is $\hat{y}_{\hat{x}_g} = C(E(\hat{x}_g, \theta_e), \theta_c)$.

Here, C depends on the correct conditional generation from D, and D depends on the classification from the network C. During the training, the C network starts to predict the category of some samples from the given labelled data correctly. Likewise, the Discriminator learns to identify the correct category for those samples and forces the D network to generate samples with the attributes related to these correctly classified samples. These generated samples bring more characteristics with them, which are not present in the given labelled data but belong to the data distribution. Now, as we feed these generated samples again to the C network with the associated conditional categories as correct labels, it learns to predict the correct category for more samples related to that generated samples. Then again, these new correctly classified samples improve the conditional generation of the D network. Hence, throughout the training, the C network and D network improve each other continuously. Meanwhile, during the training, the representation learning (latent generation) capability of the E network is also ameliorated via the process of reconstructing sample x_u, which also improves the performance of the C and D network eventually.

4) DISCRIMINATORS

The GAAE model has two discriminators: the Sample Discriminator S and the Latent Discriminator L. S makes sure that the generated sample $\hat{x_g}$ and the reconstructed sample $\hat{x_u}$ match the sample from the true data distribution p_{data}. We train S with the sample and its label. Now, for the samples $\hat{x_g}$ and $\hat{x_u}$, we have the labels y_r,$\hat{y}_{x_u}$ respectively. Hence, the pairs $(\hat{x_g}, y_r)$ and $(\hat{x_u}, \hat{y}_{x_u})$ are considered fake labels for the discriminator S. For the true data, both x_l and x_u are used together, where we get the label for the sample x_u from C, and, for the sample x_l we use the available true labels. Hence, in terms of distribution perspective, we obtain the data distribution p_{mdata}, mixing the distributions p_{ldata} and p_{data}. Accordingly, S is trained with the true sample data $x \sim p_{mdata}$ along with its associated label y if it exists, otherwise with the predicted label from C.

Here, the network E learns to map the general characteristics of the data onto the latent distribution q_z, excluding the categories from the guided data. Now, if we can draw the sample from the q_z distribution, then, by using the categorical distribution as condition, we can generate diverse data for different categories (categories from the guided data) from the Decoder D. We can only sample from q_z, if the distribution is known to us. Therefore, we use another Discriminator L so that the E network is forced to match q_z to any known distribution p_z, where p_z can be any known continuous random distribution (e. g., Continuous Normal Distribution, or Continuous Uniform Distribution). The L network is trained through differentiating between the true latent $z \sim p_z$ and the fake latent z'_{x_u}.

B. LOSSES

1) ENCODER, CLASSIFIER AND DECODER

For the E and D networks, we have the sample generation loss G_{loss}, the sample reconstruction loss R_{loss}, and the latent generation loss L_{loss}. To calculate the generation and discrimination loss, we use hinge loss, and for the reconstruction loss the Mean Squared Error (MSE) loss. For the G_{loss}, we take the average of the generation loss for $\hat{x_u}$ and $\hat{x_g}$. Therefore,

$$G_{loss} = -\frac{1}{2}(S(\hat{x_u}, \hat{y}_{x_u}; \theta_s) + S(\hat{x_g}, y_r); \theta_s). \quad (1)$$

$$L_{loss} = -(L(z'_{x_u}; \theta_l). \quad (2)$$

$$R_{loss} = \frac{1}{N}\sum_{i=1}^{N}(\hat{x}_{u_i} - x_{u_i})^2. \quad (3)$$

Now, for the C network, we calculate the classification loss Cl_{loss}, Cg_{loss} for the labelled data sample x_l and the generated sample $\hat{x_g}$ respectively. Here, $\hat{x_g}$ is used as a constant, so it is considered like a sample data x_l. We only forward propagate x_u through E and D and no gradient is calculated for generating $\hat{x_g}$ when it is only used for the loss Cg_{loss}. The model is implemented with pytorch [62] and we detach the gradient of x_g when Cg_{loss} is calculated. Therefore,

$$Cl_{loss} = -\sum y_l \log \hat{y}_{x_l}. \quad (4)$$

$$Cg_{loss} = -\sum y_r \log \hat{y}_{\hat{x}_g}. \quad (5)$$

We get the a combined loss EDC_{loss} for E,D and C. The EDC_{loss} is calculated as

$$EDC_{loss} = \alpha \cdot (\omega_1 \cdot G_{loss} + \omega_2 \cdot (\lambda \cdot R_{loss})) + \beta \cdot (\omega_3 \cdot Cl_{loss} + \omega_4 \cdot Cg_{loss} + \omega_5 \cdot L_{loss}). \quad (6)$$

Here, the weights of the E,C, and D networks are updated to minimise the loss EDC_{loss}, where $\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \alpha, \beta$, and λ are the hyperparameters. The successful training of our GAEE model depends on these parameters. At the beginning of the training, we noticed that the value of R_{loss} falls rapidly compared to other losses and results in a very small gradient value. To mitigate this problem, we multiply R_{loss} with a hyperparameter $\lambda \in \mathbb{R}_{>0}$ and after hyperparameter tuning, we found 20 as an optimal value for λ. The D network of the model is tuned for both the reconstruction loss R_loss and the generation loss G_loss. Therefore, to balance between these two losses, the hyperparameter ω_1 and ω_2 is used where ω_1, $\omega_2 \in [0, 1]$ and $\omega_1 + \omega_2 = 1$. Here, we can force the model to focus more on either loss by increasing the hyperparameter for that particular loss. Likewise, for Cl_{loss}, Cg_{loss} and L_{loss},

we use the hyperparameters ω_3, ω_4, ω_5 respectively, where $\omega_3, \omega_4, \omega_5 \in [0, 1]$ and $\omega_3 + \omega_4 + \omega_5 = 1$. In the EDC_{loss}, G_{loss} and R_{loss} are responsible for the sample generation quality, where Cl_{loss}, Cg_{loss} and L_{loss} are responsible for the latent generation quality. So, to balance between sample generation and latent generation, we use two hyperparameters α and β, where $\alpha, \beta \in [0, 1]$, and $\alpha + \beta = 1$.

2) DISCRIMINATORS' LOSS

For the Discriminators S and L, we use hinge loss. The discrimination loss for the fake samples are averaged as we calculate the loss for both $\hat{x}_u$ and $\hat{x}_g$. Let the discrimination loss for S and L be S_{loss}, L_{loss} respectively. Therefore,

$$\begin{aligned} S_{loss} = & -min(0, -1 + S(x, C(E(x, \theta_e); \theta_c); \theta_s)) \\ & - \frac{1}{2}(min(0, -1 - S(\hat{x}_u, \hat{y}_{x_u}; \theta_s)) \\ & + min(0, -1 - S(\hat{x}_g, \hat{y}_r; \theta_s))). \end{aligned} \tag{7}$$

$$\begin{aligned} L_{loss} = & -min(0, -1 + L(z, \theta_l) \\ & - min(0, -1 - L(\hat{z}_{x_u}, \theta_l)). \end{aligned} \tag{8}$$

Here, we update the parameters θ_s and θ_l to maximise the loss S_{loss} and L_{loss} respectively. Algorithm 1 shows the training mechanism for the GAAE model.

Algorithm 1 Minibatch Stochastic Gradient Descent Training of the Proposed GAAE Model. The Discriminator Is Updated k Times in One Iteration. Here, for Our Experiment, We Use $k = 2$ for Better Convergence

for number of training iterations **do**
  for k steps **do**
    Sample the latent/noise samples $\{\boldsymbol{z}^{(1)} \ldots, \boldsymbol{z}^{(m)}\}$ from p_z, the conditions (labels) $\{\boldsymbol{y_r}^{(1)}, \ldots, \boldsymbol{y_r}^{(m)}\}$ from $Cat(y_r)$, the unlabelled data samples $\{\boldsymbol{x_u}^{(1)}, \ldots, \boldsymbol{x_u}^{(m)}\}$ from p_{data} and the labelled data samples $\{\boldsymbol{x_l}^{(1)}, \ldots, \boldsymbol{x_l}^{(m)}\}$ from p_{ldata}. Here, m is the minibatch size.
    Update the discriminator S by ascending its stochastic gradient:
$$\nabla_{\theta_s} \frac{1}{m} \sum_{i=1}^{m} \left[\boldsymbol{S_{loss}}^{(i)} \right].$$
    Update the discriminator L by ascending its stochastic gradient:
$$\nabla_{\theta_l} \frac{1}{m} \sum_{i=1}^{m} \left[\boldsymbol{L_{loss}}^{(i)} \right].$$
  end for
  Repeat step [3].
  Update the Encoder E, Decoder D, and Classifier C by descending its stochastic gradient:
$$\nabla_{\theta_e, \theta_d, \theta_c} \frac{1}{m} \sum_{i=1}^{m} \left[\boldsymbol{EDC_{loss}}^{(i)} \right].$$
end for

IV. DATA AND EVALUATION METRICS

A. DATASETS

The effectiveness of the GAAE model is evaluated on both speech and non-speech audios. For the speech audio, we chose the S09 dataset [63] and the Librispeech dataset [64]. For the non-speech audio, we use the popular Nsynth dataset [65]. The S09 dataset consists of utterances for different digit categories from zero to nine. This dataset comprises 23,000 one-second audio samples uttered by 2618 speakers, where it only contains the labels for the audio digits [63].

The Librispeech dataset is an English speech dataset with 1000 hours of audio recordings, and there are three subsets available in the Librispeech dataset containing approximately 100, 300, and 500 hours of recordings, respectively. For our work, we use the subset with 100 hours of clean recordings. In this subset, the audios are uttered by 251 speakers where 125 are female, and 126 are male [64]. For our experiment, we only apply the audios along with the gender labels of the speakers.

The Nsynth audio dataset contains 305,979 musical notes of size four seconds from ten different instruments, where the sources are either acoustic, electronic, or synthetic [65]. We use three acoustic sources: Guitar, Strings, and Mallet from the Nsynth to test the compatibility of the GAAE model for a non-speech dataset.

B. DATA PREPROCESSING

We use the audio of length one second and the sampling rate of 16kHz. For the Librispeech dataset, the one-second audio is taken randomly from any particular audio clip where for the Nsynth dataset, the first one-second is taken from any audio sample as it holds the majority of the instrument sound representation.

The audio data is converted to the log-magnitude spectrograms with the short-time Fourier Transform, and the generated log-magnitude spectrograms of the GAAE model are converted to audio using the PGHI algorithm [58]. In the rest of the paper, we refer to the log-magnitude spectrogram as the spectrogram.

To obtain the spectrogram representation of the audio we followed the procedure from this paper [66]. The short-time Fourier Transform is calculated with an overlapping Hamming window of size 512 ms, and the hopping length 128 ms. Therefore, we get the size of the spectrogram as 256×128, 1D matrix. We standardise the spectrogram with the equation $\frac{X-\mu}{\sigma}$, where X is the spectrogram, μ is the mean of the spectrogram, and σ is the standard deviation of the spectrogram. We clip the dynamic range of the spectrogram at $-r$, where, for the S09 and Librispeech dataset, we determine the suitable value of r to be 10, and for the Nsynth dataset we determine it 15. Here, the log-magnitude

spectrograms is a normal distribution and any inappropriate value of the r can make the distribution skewed, which is not appropriate for training the GAAE network. We investigate the histogram of the values combining all the log-magnitude spectrograms from the whole training dataset to determine the value of r. After the clipping, we normalise the spectrogram values between -1 and 1. The spectrogram representation of the audio is used as the input to the GAAE model, which then generates spectrograms with values between -1 and 1. We then convert these spectrograms to audios via the PGHI algorithm. In this paper we refer to these audios calculated from generated spectrograms as "generated audios."

C. MEASUREMENT METRICS

We measure the performance of the GAAE model based on the generated samples and the learnt representations. The generated samples are evaluated with the Inception Score (IS) [67] and Fréchet Inception Distance (FID) [68], [69], which have become a de-facto standard for measuring the performance of any GAN based model [70].

To evaluate the representation/latent learning, we consider classification accuracy, latent space visualisation, and latent interpolation.

1) INCEPTION SCORE (IS)

The IS score is calculated based on the pretrained Inception Network [71] trained on the ImageNet dataset [72]. The logits are calculated for the images from the bottleneck layer of the Inception Network. Then, the score is calculated using

$$\exp(\mathbb{E}_x KL(p(y|x)||p(y))). \tag{9}$$

Here, x is the image sample, KL is the Kullback-Leibler Divergence (KL-divergence) [73], $p(y|x)$ is the conditional class distribution for sample x predicted by the Inception Network, and $p(y)$ is the marginal class distribution. The IS score computes the KL-divergence between the conditional label distribution and the marginal label distribution, **where the higher value indicates good generation quality.**

2) FRÉCHET INCEPTION DISTANCE (FID)

The IS score is computed solely on the generated samples; thus, no comparison is made between the generated and real samples which is not a good measure for the samples' diversity (mode) of the generated samples. The FID score solves this problem by comparing real samples with the generated samples [70] during the score calculation. The Fréchet Inception Distance (FID) computes the Fréchet Distance [74] between two multivariate Gaussian distributions for the generated and real samples, parameterised by the mean and the covariance of the features extracted from the intermediate layer of the pretrained Inception Network. The FID score is calculated using

$$||\mu_r - \mu_g||^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}), \tag{10}$$

where, μ_r, μ_g are the means for the features of the real and generated samples, respectively, and similarly, Σ_r, Σ_g are the covariances, respectively. **A lower value of the FID score indicates good generation quality.**

The Inception Network is trained on the imagenet dataset, thus, offering reliable IS and FID scores for a related image dataset, but the spectrograms of the audios are entirely different from the imagenet samples. So, the Inception Network does not offer trustworthy scores for the audio spectrograms. Hence, instead of using the Inception model, we train a classifier network based on the audio datasets and use this trained classifier to calculate the IS and FID scores. For S09 dataset, we use the pretrained classifier released by the authors of the paper "Adversarial Audio Synthesis" [11]. For the Nsynth dataset, we train a simple Convolutional Neural Network (CNN) as the Classifier, as there was no pre-trained classifier available.

V. EXPERIMENTAL SETUP, RESULTS AND DISCUSSION

For implementing our GGAN model, we follow the network implementations, optimisation, and hyperparameters from the BigGAN paper [13]. For the optimisation, we use the Adam optimiser [75]. Learning rate of $5 \cdot 10^{-5}$ is used for the networks E, D, and C, where $2 \cdot 10^{-4}$ is the learning rate for both S and L. Details of the network architectures are given in the appendix (Architectural Details).

A. IMPACT OF LABELLED DATA FOR CONDITIONAL SAMPLE GENERATION

1) SETUP

First, we evaluate the conditional sample generation quality (measured with IS and FID score) of the GAAE model for different percentage of labelled data (1% - 5 %, 100%) as guidance.

The IS and FID scores is calculated based on the 50,000 generated samples [67] for the random latent z, and the random condition y_r. The spectrograms of the samples are generated using the Decoder D network and converted to audios. These generated audios are then used to calculate the IS and FID scores. For all the datasets, we use a continuous normal distribution of size 128 to sample the latent $z \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$. For the S09 dataset, we use the ten digit categories (0-9) as the conditions $y_r \sim Cat(y_r, K = 10, p = 0.1)$. We use the three instrument categories (1-3) as conditions $y_r \sim Cat(y_r, K = 3, p = 0.33)$ for the Nsynth dataset.

For any percentage of data used as guidance, we train the GAAE model three times. Each training takes approximately 60,000 iterations with mixed-precision [76] for the batch size 128. Each time, a dataset is sampled randomly for guidance. Rest of the data is used as unsupervised manner. We limited ourselves to three times due to having high computation time: approximately 21 hours on the two Nvidia p100 GPUs. The total computation time for the S09 and the Nsynth dataset is approximately $21 \times 3 \times 6$ (1-5%,100% data) $\times$ 2 (two datasets) = 756 hours or 31.5 days.

The results of the GAAE model are compared with a Supervised BigGAN [77] and an Unsupervised BigGAN [77]. For the S09 dataset, we take the results from the GGAN publication [14]. For the Nsynth dataset, we train these models with a similar setting as was used in the GGAN paper. To calculate the IS and FID score for the Nsynth dataset, we use our pretrained supervised CNN classifier (details in the appendix) trained on three classes: Guitar, Strings, and Mallet.

2) RESULTS AND DISCUSSIONS

The percentage of labelled training data used as guidance has a significant impact on the IS and FID score, which can be found from the table 3. The more we feed the labelled data during the training, the more we boost the performance of the GAAE model for sample generation and diversity. However, notably only with 1% labelled data, the GAAE model achieves acceptable performance. For 5 % labelled data, GAAE achieves scores close to that of using 100% labelled data. So, we compare the scores for 5% data, with other models in the literature.

TABLE 1. Comparison between the sample generation quality of the GAAE model and the other models for the S09 dataset. The generation quality is measured by IS score and FID scores.

Model Name	IS Score	FID Score
Real (Train Data) [11]	9.18 ± 0.04	-
Real (Test Data) [11]	8.01 ± 0.24	-
TiFGAN [67]	5.97	26.7
WaveGAN [11]	4.67 ± 0.01	-
SpecGAN [11]	6.03 ± 0.04	-
Supervised BigGAN	$\mathbf{7.33 \pm 0.01}$	$\mathbf{24.40 \pm 0.50}$
Unsupervised BigGAN	6.17 ± 0.20	24.72 ± 0.05
GGAN [14]	7.24 ± 0.05	25.75 ± 0.10
GAAE	7.28 ± 0.01	22.60 ± 0.07

The results for S09 dataset are summarised in Table 1. Using only 5 % labelled training data as guidance, the GAAE model achieves IS score 7.28 ± 0.01 and FID score of 22.60 ± 0.07. The IS score of GAAE is close to that produced by the supervised BigGAN model (7.33 ± 0.01) and better than other models mentioned in table 1. Even the GAAE model has outperformed the supervised BigGAN model (FID score: 24.40 ± 0.50) in terms of diverse image generation, where the GAAE has used only 5 % labelled data and the supervised BigGAN is trained with all available labelled training data.

For the Nsynth dataset, the GAAE model has achieved the IS score of 2.58 ± 0.03 and the FID score of 141.71 ± 0.32 again with 5 % labelled training data as guidance. Performance of GGAN in terms of IS score is very close to that of the supervised BigGAN (2.64 ± 0.08) and better than that of the unsupervised BigGAN (2.21 ± 0.11). The performance in terms of FID score is even better than that of the supervised BigGAN (148.30 ± 0.23). Table 2 presents the comparisons.

The decoder is trained for both reconstruction and generation of the training data. During the reconstruction, it tries to reconstruct all the training samples, which helps it to learn more modes of the data distribution than the supervised BigGAN model. Figure 3 and 2 display the spectrogram of the generated and the real samples of the Nsynth, S09 datasets, respectively. From these figures, we observe that the generated samples are visually indistinguishable from the real samples. This attests the superior generation quality of the GAAE model. This is also true when we convert these spectrograms to audios. The audios can be found at: https://bit.ly/3coz5qO.

TABLE 2. Comparison between the sample generation quality of the GAAE model and the other models for the Nsynth dataset. The generation quality is measured by IS and FID scores.

Model Name	IS Score	FID Score
Real (Train Data)	2.83 ± 0.02	-
Real (Test Data)	2.81 ± 0.12	-
Supervised BigGAN	$\mathbf{2.64 \pm 0.08}$	$\mathbf{148.30 \pm 0.23}$
Unsupervised BigGAN	2.21 ± 0.11	172.01 ± 0.15
GGAN	2.52 ± 0.06	149.23 ± 0.09
GAAE	2.58 ± 0.03	141.71 ± 0.32

B. EVALUATION OF CONDITIONAL SAMPLE GENERATION BASED ON GUIDANCE

1) SETUP

In this section, we evaluate the effectiveness of guidance for accurate conditional sample generation. It is cumbersome to check all the generation manually. Therefore, we manually check only a few audio samples. For large-scale validation, we use an approach similar to [70]. We train a simple CNN classifier with the samples generated for different random conditions/categories and use the random categories associated with the generated samples as the true labels. Then, we evaluate the CNN classifier on the test dataset based on the classification accuracy. The rationale is that if the GAAE model does not learn to generate correct samples for any given category and the generated samples do not match the training data distribution; the CNN model will not be able to achieve good accuracy on the test dataset. We compare this CNN classifier with another CNN classifier which is trained using all the available training data. For further comparisons, we train two more CNN models with the generated samples from the supervised BigGAN and the GGAN model.

2) RESULT AND DISCUSSION: MANUAL TEST

The generated samples for the S09 and Nsynth dataset are shown in figure 2 and figure 3, respectively. It is not visually evident that the model was able to generate correct samples according to the given conditions/categories. However, when we convert these spectrograms to audios, it is clear that the model is able to generate audios correctly according to the categories demonstrating the effectiveness of the guidance data to learn the specific categorical distribution of the training dataset (cf. under the above link).

3) RESULTS AND DISCUSSIONS: CNN BASED CLASSIFICATION ACCURACY

For the S09 dataset, the test data classification accuracy for the CNN model trained with all the available labelled

TABLE 3. The relationship between the percentage of the data used as guidance during the training and the sample generation quality of the GAAE model, measured with the IS and the FID score. The scores are calculated for the S09 and the Nsynth dataset.

Labelled Data	IS Score (S09)	FID Score (S09)	IS Score (Nsynth)	FID Score(Nsynth)
1%	6.94 ± 0.04	24.21 ± 0.16	2.48 ± 0.08	145.89 ± 1.32
2%	7.06 ± 0.03	23.89 ± 0.11	2.53 ± 0.07	144.21 ± 0.65
3%	7.12 ± 0.04	23.15 ± 0.10	2.56 ± 0.05	143.01 ± 0.43
4%	7.19 ± 0.02	22.91 ± 0.08	2.57 ± 0.04	142.46 ± 0.38
5%	7.28 ± 0.01	22.60 ± 0.07	2.58 ± 0.03	141.71 ± 0.32
100%	$\mathbf{7.45 \pm 0.03}$	$\mathbf{19.31 \pm 0.01}$	$\mathbf{2.67 \pm 0.02}$	$\mathbf{137.65 \pm 0.02}$

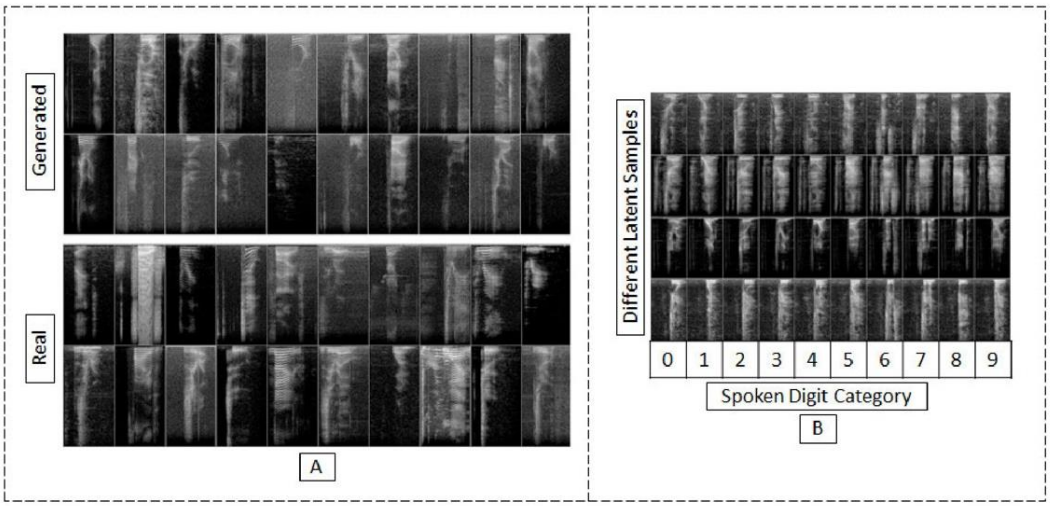

FIGURE 2. A. Illustration of the difference between the generated spectrograms and the real spectrograms of the data for the S09 dataset. The top two rows show the randomly generated samples from the GAAE model, and the bottom two rows are the real samples from the training data. Notice the visual similarity between the generated and the real samples. B. This figure shows the generated spectrograms of the S09 dataset from the GAAE model according to different digit categories. Each row represents the samples generated for a fixed latent variable where the digit condition is changed from 0 to 9. Furthermore, any column shows the generated spectrogram for a particular digit category.

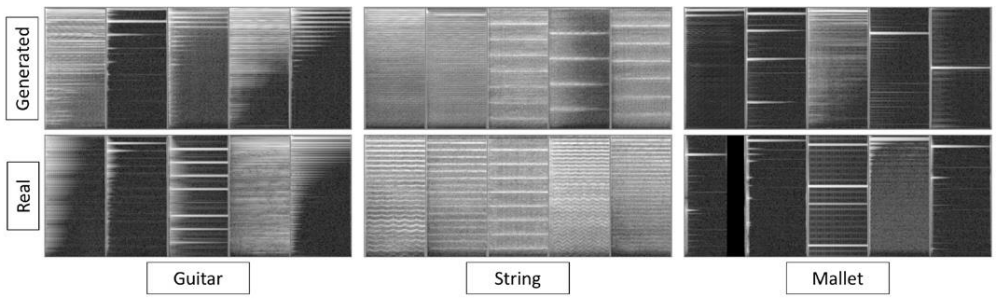

FIGURE 3. Difference between the generated spectrograms of the GAAE model and the real spectrograms of the data for the Nsynth dataset. The top row shows the generated samples, and the bottom row shows the real samples. The first block shows the spectrogram of the guitar, and the other two illustrate the spectrograms for the strings and mallet.

data is $95.52\% \pm 0.50$. The accuracy is $91.14\% \pm 0.17$, when the CNN model is trained based on the generated samples from the GAAE model (trained with 5% labelled data). The table 4 shows the comparison with other models. With the generated samples from the GAAE model, the CNN model achieves greater classification accuracy than the supervised BigGAN ($86.58\% \pm 0.56$) and the GGAN model ($86.72\% \pm 0.47$).

TABLE 4. The comparison between different CNN classifiers based on the test data classification accuracy from the S09 dataset. The CNN models are trained with the generated samples from different models.

Sample for Training	Test Accuracy
Train Data	95.52% ± 0.50
Supervised BigGAN	86.58% ± 0.56
GGAN	86.72% ± 0.47
GAAE	91.14% ± 0.17
GAAE + Train Data	**97.33% ± 0.19**

TABLE 5. The comparison between different CNN classifiers based on the test data classification accuracy from the Nsynth dataset. The CNN models are trained with the generated samples from different models.

Sample for Training	Test Accuracy
Train Data	92.01% ± 0.94
Supervised BigGAN	83.50% ± 0.62
GGAN	81.40% ± 0.48
GAAE	86.80% ± 0.23
GAAE + Train Data	**94.56% ± 0.09**

When we trained the CNN model mixing the train data, and the generated samples from the GAAE model, the accuracy of the CNN model increased from 95.52% ± 0.50 to 97.33% ± 0.19. Along with the accuracy, the stability of the CNN model is also improved significantly. This can be observed through the standard deviation in the results. We conducted the same evaluation on the Nsynth dataset and received similar results which we present in table 5.

These results demonstrate the superior performance of our GAAE model for generating samples for different categories. It can potentially be used as a data augmentation model where the generated samples from the model can be used to augment any related dataset or same dataset.

C. CONDITIONAL SAMPLE GENERATION USING GUIDANCE FROM A DIFFERENT DATASET

In the above two experiments, we used the guidance data from the same dataset. In this section, we explore the feasibility of guidance from a completely different dataset.

1) SETUP

In the S09 dataset, there are both male and female speakers, but no label is available for the gender of the speakers. We aim to verify if GAAE can generate samples from S09 dataset according to the condition on the gender category, where the guidance comes from a different dataset for gender category. To achieve this, we collect ten male and ten female speakers' audio data (randomly chosen with labels) from the Librispeech dataset to use as guidance during the training with the S09 dataset. During the training of the GAAE model, the guidance data from Librispeech dataset is also merged with S09 dataset as unlabelled data. So, GAAE learns to generate both samples from Librispeech dataset as well as from S09 dataset.

The network we used before to calculate the IS and FID score, is trained on the digit classification tasks for S09 dataset, not for the gender classification task thus will no longer offer a meaningful evaluation. To eradicate this problem, we train another simple CNN model for the gender classification to calculate the IS and the FID score. For this purpose, we randomly select 15 male and 15 female speakers from Librispeech dataset. We use data from ten male and ten female speakers for training and data from others for testing. We achieve an accuracy of 98.3 ± 0.50. We use this model to calculate the IS and FID Score for the generated samples from different models. Now, the calculated scores will reflect the quality of the generated samples according to gender distribution.

We define two GAAE models: one is trained with gender guidance, and another is trained with digit guidance. We compare the IS and FID score of these models. Note that gender information is being collected from a different dataset: Librispeech. If the gender guided model achieves better score, then we can establish the feasibility of guidance using an external dataset. To further validate this, we add results from other models (Unsupervised BigGAN, Supervised BigGAN and GGAN) trained based on digit guidance.

We choose a continuous normal distribution of size 128 for latent $z \sim \mathcal{N}(\mu = 0,\ \sigma^2 = 1)$ and two gender categories for the conditions $y_r \sim Cat(y_r, K = 2, p = 0.5)$.

2) RESULTS AND DISCUSSIONS

The calculated scores are presented in table 6. Gender guided GAAE produces the best FID and IS scores, which establish that it is feasible to get guidance from a different dataset in the GAAE model.

TABLE 6. Comparison between the performance of the GAAE model trained with gender guidance and the other models on the S09 dataset, in terms of the quality of the generated samples based on the gender attributes of the speaker, measured with the IS and the FID score.

Model Name	IS Score	FID Score
Train Data	1.92 ± 0.04	-
Test Data	1.91 ± 0.05	-
Unsupervised BigGAN	1.13 ± 0.89	56.01 ± 0.85
Supervised BigGAN	1.48 ± 0.56	35.22 ± 0.50
GGAN (Digit Guided)	1.58 ± 0.05	37.75 ± 0.10
GAAE (Digit Guided)	1.61 ± 0.17	29.84 ± 0.43
GAAE (Gender Guided)	**1.78 ± 0.03**	**20.21 ± 0.01**

D. GUIDED REPRESENTATION LEARNING

The GAAE model learns two types of representations/latent spaces: (1) it uses $z_{x_u} \sim u_z$ to learn guidance specific characteristics of the data (Guided representation) and uses (2) $z'_{x_u} \sim q_z$ to learn general characteristics of the data (General representation/Style representation).

1) SETUP

In the GAAE model, the Classifier C is built on top of the latent $z_{x_u} \sim u_z$ (see Fig. 1). The encoder network E, therefore, learns this latent variable to disentangle the class categories according to the guided data. For the S09 dataset, we use digit

classes as guidance, so, in this latent space (representation space), the digit category should be disentangled. To observe this disentanglement, we visualise the higher dimensional (128) latent space generated for the S09 test data in the 2D plane with the t-SNE (t-distributed stochastic neighbour embedding) [78] visualisation method. We use the same visualisation for the Nsynth dataset.

2) RESULTS AND DISCUSSIONS

Figure 4 shows the representation space for S09 test dataset and figure 5 shows the visualisation for the Nsynth dataset. From both figures, it is noticeable that the guided categories are well separated in the representation space, and data points of the similar categories are clustered together. So, the encoder E learns to map the data sample to the representation space u_z ensuring data categories used as guidance are well separable in the representation space.

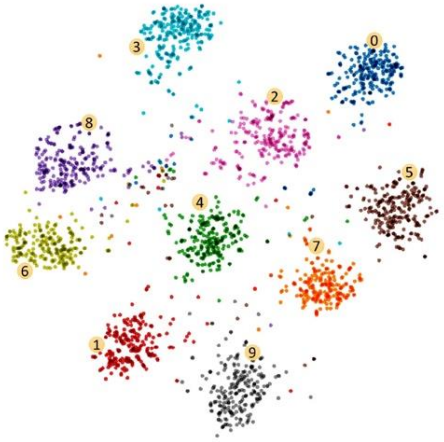

FIGURE 4. t-SNE visualisation of the learnt representation of the test data of the S09 dataset. Here, different colours of points represent different digit categories. In the representation space, the different digit categories are clustered together and easily separable.

E. GENERAL REPRESENTATION/STYLE REPRESENTATION LEARNING

1) SETUP

The encoder network E of the GAAE model is trained to match the q_z distribution with the known p_z distribution. This allows sampling z'_{x_u} from the q_z distribution.

Now, it is expected that when Decoder D learns to generate samples from the latent space q_z, it disentangles the general characteristics/attributes (independent of the guided attributes) of the data in the q_z latent space. To evaluate this disentanglement in the representation space $z'_{x_u} \sim q_z$ for both S09 and Nsynth dataset, we generate audio samples for different categories/conditions keeping the z'_{x_u} the same.

In our model, Decoder can achieve disentanglement implies that the pretrained E extracts general attributes in latent z'_{x_u} from any related dataset, which was not used during the training. To validate, we pass the test data from S09 and Nsynth dataset through E to get the general representation z'_{x_u}. Then for a fixed z'_{x_u} and different conditions (digit categories), we generate samples from the pretrained D network.

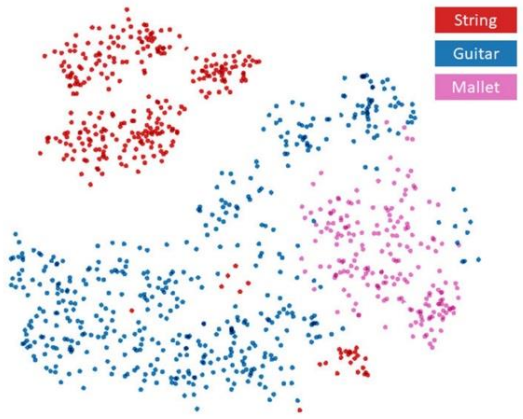

FIGURE 5. t-SNE visualisation of the learnt representation of the test data of the Nsynth dataset. Here, different colours of points represent different instrument categories. In the representation space, the different instrument categories are clustered together and easily separable.

As the GAAE model learns general/style attributes in the z'_{x_u} latent space, it should disentangle the gender of the speaker in the latent space for S09 dataset. To evaluate this, we use the trained E network from the GAAE model to extract latent representation z'_{x_u} for an entirely different Librispeech dataset where gender labels are available. For 5000 randomly sampled data from the Librispeech dataset, we extract the feature/latent z'_{x_u} from E and visualise the result in 2D plain using t-SNE visualisation for exploration.

2) RESULTS AND DISCUSSIONS

After investigating the generated audios of the S09 dataset, the digit categories are changed according to the given condition y_r and the general characteristics (such as the voice of the speaker, audio pitch, background noise etc.) of the audio is changed with the change of z'_{x_u}. So, the D network learns to capture general attributes of the data in the latent space z'_{x_u}. For the Nsynth dataset, we notice a similar behaviour.

We investigate the audio samples generated based on the extracted feature z'_{x_u} of the input data sample. Exploration of the audios shows that they preserve some characteristics (like speaker gender, voice, pitch, tone, background noise etc. for S09 test data) from the input data sample. We also notice similar scenarios for the Nsynth dataset. The audios can be found at: https://bit.ly/36Oz9z9. Note that the initial one second is the input audio data and rest are the generated audios.

Figure 6 shows the visualisation of the extracted representation for the Librispeech dataset. We observe that the

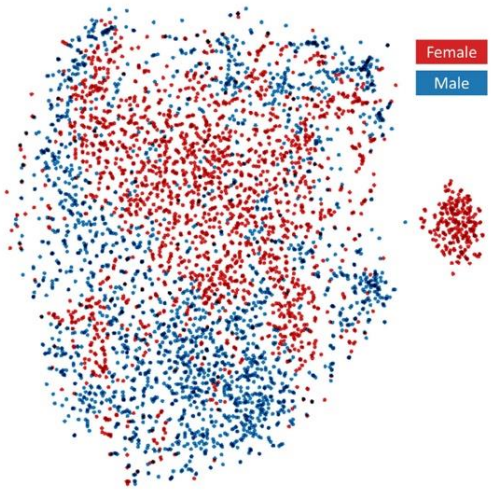

FIGURE 6. t-SNE visualisation of the learnt representation of the Libri speech dataset. Here, different colours of points represent the gender of the speakers. The representations of the different gender categories are clustered together.

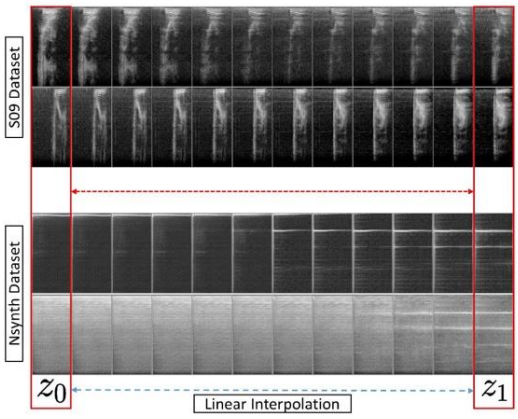

FIGURE 7. Generated spectrograms based on the linear interpolation between two latent samples; z_0 and z_1. The first two rows show the generated spectrograms for the S09 dataset (one and zero) and the bottom two rows exhibit the spectrograms for the Nsynth dataset (mallet and string). For any particular row, the first and the last spectrograms are the generations based on the fixed two latent points and the in-between spectrograms are the generation based on the interpolation between these two fixed points.

latent representation for the same gender of the speakers are clustered together and are easily separable from the latent space. This exploration exhibits that the GAAE model is able to learn the gender attributes of the speaker from the S09 dataset successfully even though gender information of the speaker was never used during the training.

F. COHERENCE OF THE GENERAL REPRESENTATION/ LATENT SPACE

1) SETUP

It is expected that the D network can learn the latent space q_z in a way so that it is coherent and if we move in any direction in the latent space the generated samples should be changed accordingly. To investigate this, we conduct linear interpolation between two latent points as described in the DCGAN paper [3]. A particular point z_i within two latent points z_0 and z_1 is calculated with the equation $z_i = z_0 + \eta(z_0 - z_1)$, where η is the step size from z_0 to z_1. With this equation, we get the latent points in between z_0 and z_1. Using this D network, we obtain the generated samples for these latent points, where the random categorical condition y_r is fixed.

2) RESULTS

Figure 7 shows the generated samples for both the S09 and Nsynth datasets based on the interpolated points. We observe that the transition between two spectrograms generated based on two fixed latent samples z_0 and z_1 is very smooth. Moreover, when we convert the spectrograms to audio, we observe the same smooth transition, which indicates the disentanglement of the general attributes in the latent space q_z The audios can be found at: https://bit.ly/2yPcTIE.

VI. HYPERPARAMETER TUNING

We tune the hyperparameters based on the S09 dataset as tuning is resource and time-intensive. We then use the hyperparameters for other datasets. From equation 6, ω_1, and ω_2 are two important hyperparameters for training the GAAE model, where $\omega_2 = 1$ - ω_1. When we increase ω_1, the model focuses more on the generation loss G_{loss} and less on the reconstruction loss R_{loss}. If we reduce ω_1, the model increases the focus for reconstruction and reduces the focus for the generation. The impact of ω_1 and ω_2 on the IS scores, FID scores, and classification accuracy are presented in figure 8. The best value for ω_1 is 0.6 and for ω_2, it is 0.4.

The α and β from equation 6 are two other important hyperparameters. The value of the α parameter determines how much the model will focus on generation (G_{loss}) and reconstruction loss (R_{loss}), where the β parameter determines the focus for the classification (Cl_{loss}, Cg_{loss}) and latent generation loss (L_{loss}). From figure 8, we observe that 0.5 is the best value for both of the hyperparameters.

There are three more hyperparameters: ω_3, ω_4, and ω_5 (See equation 6). Here, ω_3 and ω_4 control the classification loss (Cl_{loss}, Cg_{loss}) for labelled data. And, ω_5 controls the latent generation loss (L_{loss}). Here, we maintain equal balance between the classification and the latent generation loss. Likewise, we use 0.25 for ω_3,ω_4 and 0.50 for ω_5.

VII. CLASSIFIER OF THE GAAE MODEL

The success of the GAAE model is mostly dependent on its internal Classifier C. In this section, we evaluate the performance of C. We benchmark its performance using a supervised Classifier, the Classifier from GGAN and the Classifier

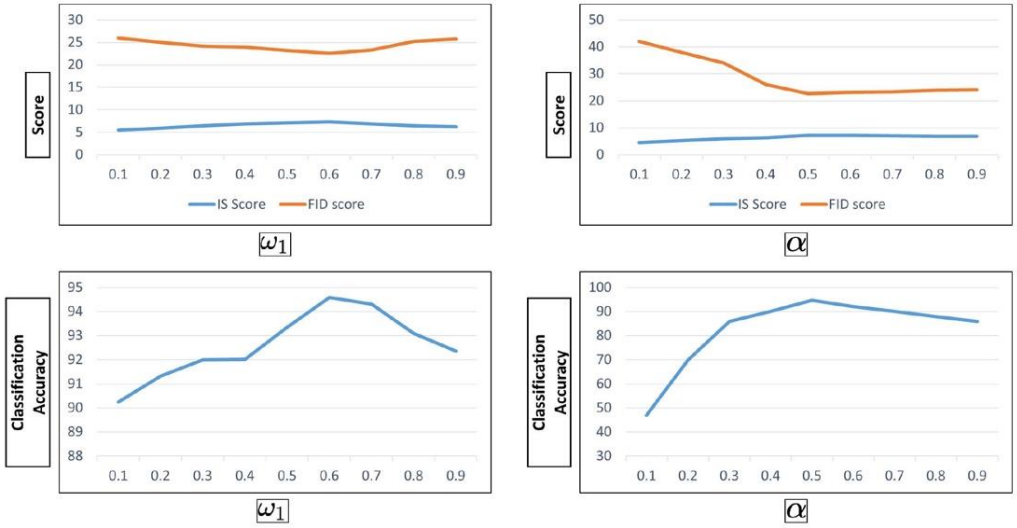

FIGURE 8. Relationship between the hyperparameters and the measurement metrics of the GAAE model. The top left plot explains the relationship between ω_1 and IS and FID scores. Similarly, the top right explicates the relationship between α and IS and FID scores. Here, The bottom left box illustrates the relationship between ω_1 and the classification accuracy. Furthermore, the bottom right plot demonstrates the impact of α on the classification accuracy.

TABLE 7. Relationship between the percentage of the data used as the guidance during the training and the S09 test dataset classification accuracy of the GAAE model.

Training Data Size	CNN Network	BiGAN	GGAN	GAAE
1%	82.21 ± 1.2	73.01 ± 1.02	84.21 ± 2.24	90.21 ± 0.16
2%	83.04 ± 0.34	75.56 ± 0.41	85.39 ± 1.24	91.45 ± 0.12
3%	83.78 ± 0.23	78.33 ± 0.07	88.25 ± 0.10	92.67 ± 0.06
4%	84.11 ± 0.34	80.03 ± 0.01	91.02 ± 0.50	93.70 ± 0.05
5%	84.50 ± 1.02	80.84 ± 1.72	92.00 ± 0.87	94.59 ± 0.03
100%	**95.52 ± 0.50**	**86.77 ± 2.61**	**96.51± 0.07**	**97.68 ± 0.01**

TABLE 8. Relationship between the percentage of the data used as the guidance during the training and the Nsynth test dataset classification accuracy of the GAAE model.

Training Data Size	CNN Network	BiGAN	GGAN	GAAE
1%	85.76 ± 1.10	82.21 ± 0.84	88.52 ± 0.32	90.26 ± 0.09
2%	89.79 ± 0.51	86.65 ± 0.57	91.69 ± 0.24	92.96 ± 0.07
3%	89.83 ± 0.49	87.21 ± 0.46	91.95 ± 0.20	93.12 ± 0.05
4%	90.52 ± 0.25	87.59 ± 0.41	92.16 ± 0.19	93.73 ± 0.02
5%	91.07 ± 0.31	87.95 ± 0.39	92.45 ± 0.14	94.23 ± 0.02
100%	**92.01 ± 0.94**	**88.09 ± 0.24**	**93.56 ± 0.09**	**94.89 ± 0.01**

from BiGAN [55]. For the supervised Classifier, we train a simple CNN classifier using 1 % - 5 %, 100 %, of training data, where the data is heavily augmented using techniques like adding random noise, rotation of the spectrogram, multiplication with random zero patches, etc. ([79]). We train a BiGAN model on top of the unsupervised BigGAN and extract BiGANs' feature network after the training. We then train another feed-forward classifier network on BiGANs' feature network using similar percentages of labelled data. We keep the weights for the feature network fixed during the training. We evaluate all these Classifiers using the test dataset. As the Classifier C of the GAAE model is trained with fewer labelled data along with the generated samples from the decoder D, it will only perform better if generation is accurate according to the different categories and the quality of the generated samples is close to the real samples.

The relationship between the percentage of the data used as guidance and the test data classification accuracy is shown in table 7, 8 for S09 and Nsyth dataset, respectively. Results from both tables demonstrate that the GAEE model

outperforms other models in terms of classification accuracy leveraging the minimal amount of labelled data (average 5%-8% percent improvement for both datasets while using 1% labelled data).

VIII. CONCLUSION AND LESSON LEARNT

In this paper, we propose the Guided Adversarial Autoencoder (GAAE), which is capable of generating high-quality audio samples using very few labelled data as guidance. After evaluating the GAAE model using two audio datasets: S09 and Nsynth, we show that the GAAE model can outperform the existing models with respect to sample generation quality and mode diversity. Harnessing the power of high-fidelity audio generation, the GAAE model can disentangle the specific attributes of the data in the learnt latent/representation space according to the guidance. This learnt representation can be beneficial to any related downstream task at hand. We also show that besides the guided representation learning, the GAAE model learns to disentangle other attributes of the data independent of the given guidance. Hence, the GAAE model learns a representation for the specific downstream task at hand and a generalised representation for future unknown related tasks.

We evaluate the GAAE model based on the audio of size one second; thus, it remains a challenge to make this model work for longer audio sample generation. In representation learning, the GAAE model can be used efficiently for any long audio sample by dividing it into one-second chunks. GAAE model successfully learns generation and representation using a minimum of 1 % labelled data. We believe this will encourage other researchers to explore the GAAE model further for few-shot learning.

Furthermore, we built the GAAE model based on BigGAN architecture. This leaves an excellent opportunity for studying other high performing GAN architectures such as progressive GAN [80] or the Style GAN [9].

APPENDIX

ARCHITECTURAL DETAILS

This section presents the details of the neural networks used in this paper. We follow the abbreviations and description style from the original work of Mario *et al.* [15].

A. SUPERVISED BigGAN

We use the exact implementation of the Supervised BigGAN from our former GGAN paper [14]. Therefore, for the implementation of both the Generator and the Discriminator, we apply a Resnet architecture from the BigGAN work [13]. The layers are shown tables 10 and 11. The Generator and Discriminator architectures are shown in Tables 12 and 13, respectively. We use a learning rate of 0.00005 and 0.0002 for the Generator and the Discriminator, respectively. We set the number of channels (ch) to 16 to minimise the computational expenses, as the higher number of channels such as 64 and 32 only offer negligible improvements.

TABLE 9. Abbreviations for defining the architectures.

Full Name	Abbreviation
Resample	RS
Batch normalisation	BN
Conditional batch normalisation	cBN
Downscale	D
Upscale	U
Spectral normalisation	SN
Input height	h
Input width	w
True label	y
Input channels	ci
Output channels	co
Number of channels	ch

TABLE 10. Architecture of the ResBlock generator with upsampling for the supervised BigGAN.

Layer Name	Kernal Size	RS	Output Size
Shortcut	[1,1,1]	U	2h × 2w × c_{o}
cBN, ReLU	-	-	h × w × c_{i}
Convolution	[3,3,1]	U	2h × 2w × c_{o}
cBN, ReLU	-	-	2h × 2w × c_{o}
Convolution	[3,3,1]	U	2h × 2w × c_{o}
Addition	-	-	2h × 2w × c_{o}

TABLE 11. Architecture of the ResBlock discriminator with downsampling for the supervised BigGAN.

Layer Name	Kernal Size	RS	Output Size
Shortcut	[1,1,1]	D	h/2 × w/2 × c_{o}
ReLU	-	-	h × w × c_{i}
Convolution	[3,3,1]	-	h × w × c_{o}
ReLU	-	-	h × w × c_{o}
Convolution	[3,3,1]	D	h/2 × w/2 × c_{o}
Addition	-	-	h/2 × w/2 × c_{o}

TABLE 12. Architecture of the generator for the supervised BigGAN.

Layer Name	RS	SN	Output Size
Input z	-	-	128
Dense	-	-	4 × 2 × 16. ch
ResBlock	U	SN	8 × 4 × 16. ch
ResBlock	U	SN	16 × 8 × 16. ch
ResBlock	U	SN	32 × 16 × 16. ch
ResBlock	U	SN	64 × 32 × 16. ch
ResBlock	U	SN	128 × 64 × 16. ch
Non-local block	-	-	128 × 64 × 16. ch
ResBlock	U	SN	256 × 128 × 1. ch
BN, ReLU	-	-	256 × 128 × 1
Conv [3, 3, 1]	-	-	256 × 128 × 1
Tanh	-	-	256 × 128 × 1

B. UNSUPERVISED BigGAN

Similarly, for the unsupervised BigGAN, follow the same implementation from the original GGAN work [14]. Tables 14 and 15 show the upsampling and downsampling layers, respectively. The architectures of the Generator and Discriminator are shown in the tables 16 and 17, respectively.

TABLE 13. Architecture of the discriminator for the supervised BigGAN.

Layer Name	RS	Output Size
Input Spectrogram	-	256 × 128 × 1
ResBlock	D	128 × 64 × 1. ch
Non-local block	-	128 × 64 × 1. ch
ResBlock	-	64 × 32 × 1. ch
ResBlock	D	32 × 16 × 2. ch
ResBlock	D	16 × 8 × 4. ch
ResBlock	D	8 × 4 × 8. ch
ResBlock	D	4 × 2 × 16. ch
ResBlock (No Shortcut)	-	4 × 2 × 16. ch
ReLU	-	4 × 2 × 16. ch
Global sum pooling	-	1 × 1 × 16. ch
Sum(embed(y)·h)+(dense → 1)	-	1

TABLE 14. Architecture of the ResBlock generator with upsampling for the unsupervised BigGAN.

Layer Name	Kernal Size	RS	Output Size
Shortcut	[1,1,1]	U	2h × 2w × c_{o}
BN, ReLU	-	-	h × w × c_{i}
Convolution	[3,3,1]	U	2h × 2w × c_{o}
BN, ReLU	-	-	2h × 2w × c_{o}
Convolution	[3,3,1]	U	2h × 2w × c_{o}
Addition	-	-	2h × 2w × c_{o}

The learning rate and channels are the same as for the supervised BigGAN.

C. BiGAN

For the BiGAN model, we train a Feature Extractor and Discriminator network on top of the unsupervised BigGAN. The Feature Extractor network creates the features for real samples, and the Discriminator tries to differentiate between the generated features and the random noise. The detail is exactly followed from the original BiGAN work [55]. The downsampling layer is the same as the unsupervised BigGAN and can be found in table 15. The architecture of the Feature Extractor network is shown in table 18. Furthermore, the architecture of the Discriminator is given in table 19.

TABLE 15. Architecture of the ResBlock discriminator with downsampling for the unsupervised BigGAN.

Layer Name	Kernal Size	RS	Output Size
Shortcut	[1,1,1]	D	h/2 × w/2 × c_{o}
ReLU	-	-	h × w × c_{i}
Convolution	[3,3,1]	-	h × w × c_{o}
ReLU	-	-	h × w × c_{o}
Convolution	[3,3,1]	D	h/2 × w/2 × c_{o}
Addition	-	-	h/2 × w/2 × c_{o}

D. GAAE

In the GAAE model, the downsampling and upsampling layers are the same as those shown in table 10 and 11, respectively.

TABLE 16. Architecture of the generator for the unsupervised BigGAN.

Layer Name	RS	SN	Output Size
Input z	-	-	128
Dense	-	-	4 × 2 × 16. ch
ResBlock	U	SN	8 × 4 × 16. ch
ResBlock	U	SN	16 × 8 × 16. ch
ResBlock	U	SN	32 × 16 × 16. ch
ResBlock	U	SN	64 × 32 × 16. ch
ResBlock	U	SN	128 × 64 × 16. ch
Non-local block	-	-	128 × 64 × 16. ch
ResBlock	U	SN	256 × 128 × 1. ch
BN, ReLU	-	-	256 × 128 × 1
Conv [3, 3, 1]	-	-	256 × 128 × 1
Tanh	-	-	256 × 128 × 1

TABLE 17. Architecture of the discriminator for the unsupervised BigGAN.

Layer Name	RS	Output Size
Input Spectrogram	-	256 × 128 × 1
ResBlock	D	128 × 64 × 1. ch
Non-local block	-	128 × 64 × 1. ch
ResBlock	-	64 × 32 × 1. ch
ResBlock	D	32 × 16 × 2. ch
ResBlock	D	16 × 8 × 4. ch
ResBlock	D	8 × 4 × 8. ch
ResBlock	D	4 × 2 × 16. ch
ResBlock (No Shortcut)	-	4 × 2 × 16. ch
ReLU	-	4 × 2 × 16. ch
Global sum pooling	-	1 × 1 × 16. ch
Dense	-	1

TABLE 18. Architecture of the Feature Extractor Network for the BiGAN.

Layer Name	RS	Output Size
Input Spectrogram	-	256 × 128 × 1
ResBlock	D	128 × 64 × 1. ch
Non-local block	-	128 × 64 × 1. ch
ResBlock	-	64 × 32 × 1. ch
ResBlock	D	32 × 16 × 2. ch
ResBlock	D	16 × 8 × 4. ch
ResBlock	D	8 × 4 × 8. ch
ResBlock	D	4 × 2 × 16. ch
ResBlock (No Shortcut)	-	4 × 2 × 16. ch
ReLU	-	4 × 2 × 16. ch
Global sum pooling	-	1 × 1 × 16. ch
Dense	-	128

The Encoder architecture is given in table 20, where we use two dense layers to obtain z_{x_u} and z'_{x_u} from a global sum pooling layer. For the Decoder, the conditional vector y_r or $\hat{y}_{x_u}$ is given through the conditional Batch Normaliser (cBN) from the upsampling layer. The classifier network is built upon some dense layer, and the architecture is given in table 22. For the Sample Discriminator, we exactly follow the implementation in table 13. Here, in the table 13, y is the conditional vector, and h is the output from the global sum pooling layer. For the Latent Discriminator, we have

TABLE 19. Architecture of the Discriminator for the BiGAN.

Layer Name	RS	Output Size
Input Spectrogram	-	256 × 128 × 1
ResBlock	D	128 × 64 × 1. ch
Non-local block	-	128 × 64 × 1. ch
ResBlock	-	64 × 32 × 1. ch
ResBlock	D	32 × 16 × 2. ch
ResBlock	D	16 × 8 × 4. ch
ResBlock	D	8 × 4 × 8. ch
ResBlock	D	4 × 2 × 16. ch
ResBlock (No Shortcut)	-	4 × 2 × 16. ch
ReLU	-	4 × 2 × 16. ch
Global sum pooling	-	1 × 1 × 16. ch
Concat with input feature	-	256+128=384
Dense	-	128
ReLU	-	128
Dense	-	1

TABLE 20. Architecture of the Encoder for the GAAE.

Layer Name	RS	Output Size
Input Spectrogram	-	256 × 128 × 1
ResBlock	D	128 × 64 × 1. ch
Non-local block	-	128 × 64 × 1. ch
ResBlock	-	64 × 32 × 1. ch
ResBlock	D	32 × 16 × 2. ch
ResBlock	D	16 × 8 × 4. ch
ResBlock	D	8 × 4 × 8. ch
ResBlock	D	4 × 2 × 16. ch
ResBlock (No Shortcut)	-	4 × 2 × 16. ch
ReLU	-	4 × 2 × 16. ch
Global sum pooling	-	1 × 1 × 16. ch
Dense (z_{x_u}), Dense (z'_{x_u})	-	128, 128

TABLE 21. Architecture of the Decoder for the GAAE.

Layer Name	RS	SN	Output Size
Input latent vector	-	-	128
Dense	-	-	4 × 2 × 16. ch
ResBlock	U	SN	8 × 4 × 16. ch
ResBlock	U	SN	16 × 8 × 16. ch
ResBlock	U	SN	32 × 16 × 16. ch
ResBlock	U	SN	64 × 32 × 16. ch
ResBlock	U	SN	128 × 64 × 16. ch
Non-local block	-	-	128 × 64 × 16. ch
ResBlock	U	SN	256 × 128 × 1. ch
BN, ReLU	-	-	256 × 128 × 1
Conv [3, 3, 1]	-	-	256 × 128 × 1
Tanh	-	-	256 × 128 × 1

use multi dense layers, and the architecture is given in table 23.

The learning rates for both Discriminators are 0.0002, and for other networks, the learning rate is 0.00005. We set the number of channels to 16 for all the experiment carried out with the GAAE.

TABLE 22. Architecture of the Classifier for the GGAN.

Layer Name	Output Size
Input latent vector	128
Dense	128
ReLU	128
Dense	10

TABLE 23. Architecture of the Latent Discriminator for the GGAN.

Layer Name	Output Size
Input latent vector	128
Dense	128
ReLU	128
Dense	128
ReLU	128
Dense	1

TABLE 24. Architecture of the Simple Spectrogram Classifier.

Layer Name	Output Size
Input Spectrogram	256 × 128 × 1
Convolution [3, 3, 32]	256 × 128 × 32
Maxpool [2, 2]	128 × 64 × 32
Convolution [3, 3, 64]	128 × 64 × 64
Maxpool [2, 2]	64 × 32 × 64
Convolution [3, 3, 128]	64 × 32 × 128
Maxpool [2, 2]	32 × 16 × 128
Convolution [3, 3, 256]	32 × 16 × 256
Maxpool [2, 2]	16 × 8 × 256
Dense	c

E. SIMPLE CLASSIFIER

For many classification tasks, we mention a Simple Classifier throughout the paper. The architecture of these classifiers are as in table 24. Here, c is the number of outputs according to the classification categories. The learning rates is used as 0.0001 for this classifier network.

REFERENCES

[1] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. F. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4114–4124.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014, pp. 2672–2680.

[3] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, pp. 1–16, Nov. 2015.

[4] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[5] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using WaveNet autoencoders," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 12, pp. 2041–2053, Dec. 2019.

[6] J. Donahue and K. Simonyan, "Large scale adversarial representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10541–10551.

[7] S. Zhao, J. Song, and S. Ermon, "InfoVAE: Information maximizing variational autoencoders," 2017, *arXiv:1706.02262*. [Online]. Available: http://arxiv.org/abs/1706.02262
[8] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*. [Online]. Available: http://arxiv.org/abs/1511.05644
[9] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4401–4410.
[10] J. Engel, K. Krishna Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial neural audio synthesis," 2019, *arXiv:1902.08710*. [Online]. Available: http://arxiv.org/abs/1902.08710
[11] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–16.
[12] A. Marafioti, N. Holighaus, N. Perraudin, and P. Majdak, "Adversarial generation of time-frequency features with application in audio synthesis," 2019, *arXiv:1902.04072*. [Online]. Available: http://arxiv.org/abs/1902.04072
[13] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," *CoRR*, vol. abs/1809.11096, pp. 1–35, Sep. 2018.
[14] K. Nazmul Haque, R. Rana, J. H. L. Hansen, and B. Schuller, "Guided generative adversarial neural network for representation learning and high fidelity audio generation using fewer labelled audio data," 2020, *arXiv:2003.02836*. [Online]. Available: http://arxiv.org/abs/2003.02836
[15] M. Lucic, M. Tschannen, M. Ritter, X. Zhai, O. Bachem, and S. Gelly, "High-fidelity image generation with fewer labels," 2019, *arXiv:1903.02271*. [Online]. Available: http://arxiv.org/abs/1903.02271
[16] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, New York, NY, USA, vol. 1. Red Hook, NY, USA: Curran Associates, 2013, pp. 899–907.
[17] R. Zhang, P. Isola, and A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9907, Oct. 2016, pp. 649–666.
[18] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 577–593.
[19] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1422–1430.
[20] K. Nazmul Haque, M. Abu Yousuf, and R. Rana, "Image denoising and restoration with CNN-LSTM encoder decoder with direct attention," 2018, *arXiv:1801.05141*. [Online]. Available: http://arxiv.org/abs/1801.05141
[21] X. Zhan, X. Pan, Z. Liu, D. Lin, and C. C. Loy, "Self-supervised learning via conditional motion propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1881–1889.
[22] Z. Feng, C. Xu, and D. Tao, "Self-supervised representation learning by rotation feature decoupling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10364–10374.
[23] S. Liu, A. Davison, and E. Johns, "Self-supervised generalisation with meta auxiliary learning," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1679–1689, 2019.
[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[25] J. Wu, X. Wang, and W. Yang Wang, "Self-supervised dialogue learning," 2019, *arXiv:1907.00448*. [Online]. Available: http://arxiv.org/abs/1907.00448
[26] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "VL-BERT: Pre-training of generic visual-linguistic representations," 2019, *arXiv:1908.08530*. [Online]. Available: http://arxiv.org/abs/1908.08530
[27] H. Wang, X. Wang, W. Xiong, M. Yu, X. Guo, S. Chang, and W. Y. Wang, "Self-supervised learning for contextualized extractive summarization," 2019, *arXiv:1906.04466*. [Online]. Available: http://arxiv.org/abs/1906.04466
[28] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *CoRR*, vol. abs/1803.07728, pp. 1–16, Oct. 2018.
[29] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*. [Online]. Available: http://arxiv.org/abs/1807.03748
[30] F. de Chaumont Quitry, M. Tagliasacchi, and D. Roblek, "Learning audio representations via phase prediction," 2019, *arXiv:1910.11910*. [Online]. Available: http://arxiv.org/abs/1910.11910
[31] A. Nagrani, J. S. Chung, S. Albanie, and A. Zisserman, "Disentangled speech embeddings using cross-modal self-supervision," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6829–6833.
[32] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition," 2019, *arXiv:1904.05862*. [Online]. Available: http://arxiv.org/abs/1904.05862
[33] Y.-A. Chung and J. Glass, "Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech," 2018, *arXiv:1803.08976*. [Online]. Available: http://arxiv.org/abs/1803.08976
[34] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Audio Word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," 2016, *arXiv:1603.00982*. [Online]. Available: http://arxiv.org/abs/1603.00982
[35] K. Kawakami, L. Wang, C. Dyer, P. Blunsom, and A. van den Oord, "Learning robust and multilingual speech representations," 2020, *arXiv:2001.11128*. [Online]. Available: http://arxiv.org/abs/2001.11128
[36] M. Riviere, A. Joulin, P.-E. Mazare, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 7414–7418.
[37] A. Baevski, S. Schneider, and M. Auli, "Vq-wav2vec: Self-supervised learning of discrete speech representations," 2019, *arXiv:1910.05453*. [Online]. Available: http://arxiv.org/abs/1910.05453
[38] A. Baevski, M. Auli, and A. Mohamed, "Effectiveness of self-supervised pre-training for speech recognition," 2019, *arXiv:1911.03912*. [Online]. Available: http://arxiv.org/abs/1911.03912
[39] S. Latif, R. Rana, S. Khalifa, R. Jurdak, J. Qadir, and B. W. Schuller, "Deep representation learning in speech processing: Challenges, recent advances, and future trends," 2020, *arXiv:2001.00378*. [Online]. Available: http://arxiv.org/abs/2001.00378
[40] S. Amiriparian, M. Freitag, N. Cummins, and B. Schuller, "Sequence to sequence autoencoders for unsupervised representation learning from audio," in *Proc. Workshop (DCASE)*, 2017, pp. 1–5.
[41] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1096–1104.
[42] Y. Xu, Q. Huang, W. Wang, P. Foster, S. Sigtia, P. J. B. Jackson, and M. D. Plumbley, "Unsupervised feature learning based on deep models for environmental audio tagging," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 6, pp. 1230–1241, Jun. 2017.
[43] M. Neumann and N. T. Vu, "Improving speech emotion recognition with unsupervised representation learning on unlabeled speech," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 7390–7394.
[44] S. Ghosh, E. Laksana, L.-P. Morency, and S. Scherer, "Learning representations of affect from speech," 2015, *arXiv:1511.04747*. [Online]. Available: http://arxiv.org/abs/1511.04747
[45] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2017, pp. 16–23.
[46] W.-N. Hsu, Y. Zhang, R. J. Weiss, Y.-A. Chung, Y. Wang, Y. Wu, and J. Glass, "Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 5901–5905.
[47] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114
[48] J. Chang and S. Scherer, "Learning representations of emotional speech with deep convolutional generative adversarial networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2746–2750.
[49] H. Yu, Z.-H. Tan, Z. Ma, and J. Guo, "Adversarial network bottleneck features for noise robust speaker verification," 2017, *arXiv:1706.03397*. [Online]. Available: http://arxiv.org/abs/1706.03397
[50] S. Sahu, R. Gupta, G. Sivaraman, W. AbdAlmageed, and C. Espy-Wilson, "Adversarial auto-encoders for speech based emotion recognition," 2018, *arXiv:1806.02146*. [Online]. Available: http://arxiv.org/abs/1806.02146
[51] E. Principi, F. Vesperini, S. Squartini, and F. Piazza, "Acoustic novelty detection with adversarial autoencoders," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 3324–3330.
[52] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. C. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *Proc. ICLR*, 2017, pp. 1–6.

[53] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards End-to-End speech synthesis," 2017, *arXiv:1703.10135*. [Online]. Available: http://arxiv.org/abs/1703.10135

[54] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," 2016, *arXiv:1606.00704*. [Online]. Available: http://arxiv.org/abs/1606.00704

[55] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *CoRR*, vol. abs/1605.09782, pp. 1–18, May 2016.

[56] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8110–8119.

[57] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 2, pp. 236–243, Apr. 1984.

[58] Z. Průša, P. L. Søndergaard, N. Holighaus, C. Wiesmeyr, and P. Balazs, "The large time-frequency analysis toolbox 2.0," in *Sound, Music, and Motion*, M. Aramaki, O. Derrien, R. Kronland-Martinet, and S. Ystad, Eds. Cham, Switzerland: Springer, 2014, pp. 419–442.

[59] A. Spurr, E. Aksan, and O. Hilliges, "Guiding InfoGAN with semi-supervision," in *Machine Learning and Knowledge Discovery in Databases*, M. Ceci, J. Hollmén, L. Todorovski, C. Vens, and S. Džeroski, Eds. Cham, Switzerland: Springer, 2017, pp. 119–134.

[60] J. Tobias Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," 2015, *arXiv:1511.06390*. [Online]. Available: http://arxiv.org/abs/1511.06390

[61] K. Sricharan, R. Bala, M. Shreve, H. Ding, K. Saketh, and J. Sun, "Semi-supervised conditional GANs," 2017, *arXiv:1708.05789*. [Online]. Available: http://arxiv.org/abs/1708.05789

[62] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[63] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *CoRR*, vol. abs/1804.03209, pp. 1–11, Apr. 2018.

[64] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

[65] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2017, pp. 1068–1077.

[66] A. Marafioti, N. Holighaus, N. Perraudin, and P. Majdak, "Adversarial generation of time-frequency features with application in audio synthesis," *CoRR*, vol. abs/1902.04072, pp. 4352–4356, May 2019.

[67] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016, pp. 2234–2242.

[68] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local NASH equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.

[69] S. Barratt and R. Sharma, "A note on the inception score," 2018, *arXiv:1801.01973*. [Online]. Available: http://arxiv.org/abs/1801.01973

[70] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my GAN?" in *Computing Vision—ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 218–234.

[71] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[72] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[73] T. van Erven and P. Harremos, "Rényi divergence and Kullback-Leibler divergence," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 3797–3820, Jul. 2014.

[74] D. C. Dowson and B. V. Landau, "The Fréchet distance between multivariate normal distributions," *J. Multivariate Anal.*, vol. 12, no. 3, pp. 450–455, Sep. 1982.

[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, pp. 1–15, Dec. 2015.

[76] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, "Mixed precision training," 2017, *arXiv:1710.03740*. [Online]. Available: http://arxiv.org/abs/1710.03740

[77] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*. [Online]. Available: http://arxiv.org/abs/1809.11096

[78] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[79] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," 2019, *arXiv:1904.08779*. [Online]. Available: http://arxiv.org/abs/1904.08779

[80] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*. [Online]. Available: http://arxiv.org/abs/1710.10196

KAZI NAZMUL HAQUE received the master's degree in information technology from Jahangirnagar University, Bangladesh. He is currently pursuing the Ph.D. degree with the University of Southern Queensland, Australia. He has been working professionally in the field of machine learning for more than five years. His research interest includes building machine learning models to solve diverse real-world problems. The current focus of his research work is unsupervised representation learning for audio data.

RAJIB RANA (Member, IEEE) received the B.Sc. degree in computer science and engineering from Khulna University, Bangladesh, with the Prime Minister and President's Gold Medal for outstanding achievements, and the Ph.D. degree in computer science and engineering from the University of New South Wales, Sydney, Australia, in 2011. He received his Postdoctoral Training with the Autonomous Systems Laboratory, CSIRO, before joining the University of Southern Queensland, as a Faculty Member, in 2015. He is currently an Experimental Computer Scientist, an Advance Queensland Research Fellow, and a Senior Lecturer with the University of Southern Queensland. He is also the Director of the IoT Health Research Program with the University of Southern Queensland. His research work aims to capitalize on advancements in technology along with sophisticated information and data processing to better understand disease progression in chronic health conditions and develop predictive algorithms for chronic diseases, such as mental illness and cancer. His current research interest includes unsupervised representation learning. He was a recipient of the Prestigious Young Tall Poppy QLD Award, in 2018, as one of the Queensland's most outstanding scientists for achievements in the area of scientific research and communication.

BJÖRN W. SCHULLER, JR. (Fellow, IEEE) received the diploma degree, the Ph.D. degree in automatic speech and emotion recognition, and the habilitation and an Adjunct Teaching Professorship in signal processing and machine intelligence from Technische Universität München (TUM), Munich, Germany, in 1999, 2006, and 2012, respectively, all in electrical engineering and information technology. He is currently a Professor of Artificial Intelligence with the Department of Computing, Imperial College London, U.K., where he heads the Group on Language, Audio and Music (GLAM), a Full Professor and the Head of the Chair of Embedded Intelligence for Health Care and Wellbeing with the University of Augsburg, Germany, and a Founding CEO/CSO of audEERING. He was previously a Full Professor and the Head of the Chair of Complex and Intelligent Systems with the University of Passau, Germany. He has (co-)authored five books and more than 900 publications in peer-reviewed books, journals, and conference proceedings leading to more than overall 30 000 citations (H-index=82). He was an Elected Member of the IEEE Speech and Language Processing Technical Committee. He is a Golden Core Member of the IEEE Computer Society, a Fellow of the ISCA, a Senior Member of the ACM, and the President-Emeritus of the Association for the Advancement of Affective Computing (AAAC). He was the General Chair of ACII 2019, a Co-Program Chair of Interspeech, in 2019, and ICMI, in 2019, a repeated Area Chair of ICASSP, next to a multitude of further Associate and a Guest Editor roles and functions in Technical and Organisational Committees. He is the Field Chief Editor of the *Frontiers in Digital Health* and a former Editor in Chief of the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING.

4.3. Links and implications

In this chapter, I have demonstrated the capabilities of the Guided Adversarial Autoencoder (GAAE) model to simultaneously acquire both guided and generalised representations from unlabelled audio data. Guided representation learning enhances performance in related tasks where labelled audio data is limited, while the generalised representation can be applied to entirely unrelated tasks. This research introduces a significant opportunity for researchers to harness unlabelled audio data for enhancing various audio-related tasks.

However, it's worth noting that both the proposed GGAN and GAAE models rely on manually extracted spectrograms, which may limit their full potential. These models are built upon a Convolutional Neural Network (CNN) architecture. To address this limitation and reduce the dependency on spectrograms, I have presented an enhanced version of the CNN model in the next chapter. This improved model is designed to directly model audio from raw waveforms and holds the potential for integration within the GAAE and GGAN frameworks.

CHAPTER 5: PAPER 3 – Raw Audio Classification with Cosine Convolutional Neural Network (CosCovNN)

5.1. Introduction

In this chapter, I introduce the Cosine Convolutional Neural Network (CosCovNN) as an innovative alternative to the traditional CNN model for the direct classification of audio from raw waveforms. CosCovNN offers the potential to replace the core CNN architecture of GGAN and GAAE, thereby eliminating their reliance on handcrafted features. This development aligns precisely with the third objective of the research work.

CosCovNN can significantly reduce the parameter count by 77%, while consistently outperforming similar CNN models in audio classification tasks across five different datasets. While CosCovNN excels, it does not surpass the performance of complex models found in the existing literature. To further emphasise the potential of CosCovNN, I propose the Vector Quantised Cosine Convolutional Neural Network with Memory (VQCCM). Through rigorous evaluation and benchmarking against existing literature, I demonstrate that VQCCM achieves state-of-the-art performance across various audio classification tasks, often surpassing the performance of existing models found in the literature.

5.2. Published paper

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier XXXXXX/ACCESS.XXXX.DOI

Raw Audio Classification with Cosine Convolutional Neural Network (CosCovNN)

KAZI NAZMUL HAQUE[1], RAJIB RANA[1], AND BJÖRN W. SCHULLER, JR.[2,3]
[1]University of Southern Queensland, Australia
[2]GLAM – Group on Language, Audio, & Music, Imperial College London, UK
[3]Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg, Germany

Corresponding author: Kazi Nazmul Haque (e-mail: shezan.huq@gmail.com).

ABSTRACT

This study explores the field of audio classification from raw waveform using Convolutional Neural Networks (CNNs), a method that eliminates the need for extracting specialised features in the pre-processing step. Unlike recent trends in literature, which often focuses on designing frontends or filters for only the initial layers of CNNs, our research introduces the Cosine Convolutional Neural Network (CosCovNN) replacing the traditional CNN filters with Cosine filters. The CosCovNN surpasses the accuracy of the equivalent CNN architectures with approximately 77% less parameters. Our research further progresses with the development of an augmented CosCovNN named Vector Quantised Cosine Convolutional Neural Network with Memory (VQCCM), incorporating a memory and vector quantisation layer. VQCCM achieves state-of-the-art (SOTA) performance across five different datasets in comparison with existing literature. Our findings show that cosine filters can greatly improve the efficiency and accuracy of CNNs in raw audio classification.

INDEX TERMS Audio Classification, Convolutional Neural Network, Cosine Filter, Vector Quantisation, CNN with Memory

I. INTRODUCTION

Convolutional Neural Networks have been remarkably successful in computer vision for modelling directly from raw data, eliminating the need for handcrafted features [1]. Similarly, in audio classification, there is a growing interest in direct raw waveform modelling. This approach is particularly challenging due to the high dimensionality and complex temporal dependencies inherent in audio data, necessitating advanced and computationally robust CNN architectures [2]. Directly modelling from raw waveform eliminates preprocessing and reduces manual intervention, aligning with the data-driven characteristics of deep learning [3]. Unlike traditional spectrogram-based CNNs, which are limited to certain frequencies, CNNs processing raw waveforms can identify a wider range of frequency responses, becoming more effective as more data becomes available. [4].

Responding to the evolving field, researchers have explored various modifications of CNN architectures to effectively handle audio waveforms [5]–[8]. Much of this research has been centred on developing front-end modules and filters, particularly enhancing the initial layers of waveform-based CNNs. This is based on the understanding that the first layer is crucial as it directly interacts with the data. Notable contributions in this area include the SincNet filter by Ravanelli et al. [9], designed for the initial layers of the CNN, and Zeghidour et al.'s [10] LEAF, a learnable front-end adaptable to various neural networks. The key concept of these advancements is the replacement of handcrafted features with learnable filters or frontends in the early stages of the processing, but they still rely on traditional CNN architecture beyond the initial layers.

Building on this direction, we propose a new approach, CosCovNN, which integrates a cosine filter into the CNN framework. This filter, designed to replace traditional CNN kernels, draws its inspiration from the principles of the Discrete Cosine Transform (DCT) and the real parts of the Fourier Transform. The DCT, known for its ability to represent signals through a summation of cosine functions at various frequencies, efficiently captures the spectral characteristics inherent in audio signals [11]–[13]. At the same

time, the real parts of the Fourier Transform, which are made up of cosine elements, play a crucial role in highlighting the symmetrical parts of a signal's frequency range. These elements are fundamental in audio processing, particularly in identifying rhythmic and harmonic structures [14]–[16]. Our choice of a cosine-based filter aligns with traditional signal processing methods, enhancing the CNN's ability to interpret complex audio patterns.

To enhance CosCovNN's capabilities, we've added a Vector Quantisation layer after the first convolutional layer, encouraging the model to focus on extracting significant features from audio waveform [17]. Moreover, we have added a memory module which allows the initial layers of the network to propagate important information of the data directly to the final layers improving its performance in the task of raw audio modelling [18]–[20].

In this paper, we make the following key contributions:

- The development of CosCovNN (Cosine Convolutional Neural Network), a novel CNN based architecture that introduces learnable cosine filters. CosCovNN not only surpasses traditional CNN models in performance but also achieves this with approximately 77% fewer parameters.
- The introduction of the Vector Quantised Cosine Convolutional Neural Network with Memory (VQCCM), an advanced version of CosCovNN. We conduct thorough evaluations of both CosCovNN and VQCCM for classification tasks using five different datasets. This allows us to compare our results directly with recent studies in the field. The VQCCM achieves state-of-the-art (SOTA) performance and sets new benchmarks in several cases. This highlights the effectiveness and potential of our cosine filter-based CNN models in the field of raw audio classification.

II. BACKGROUND AND RELATED WORK

The domain of audio classification has been revolutionised by deep learning, transitioning from traditional signal processing methods such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Hidden Markov Models (HMM) [21]–[23] to more sophisticated neural network models [24]–[26]. Early approaches relied heavily on hand-crafted features, but the advent of deep learning, particularly Convolutional Neural Networks (CNNs) [27]–[29] and Recurrent Neural Networks (RNNs) [30]–[32], ushered in a new era of audio classification. This shift was motivated by the need to capture more complex patterns in audio signals, a task at which deep learning models excel due to their ability to learn powerful representations directly from raw data [33].

Despite the advancements brought by deep learning in audio classification, a significant portion of these models still primarily rely on 2D representations of audio data, such as MFCC coefficients, FBANK, and spectrograms [34]–[37]. However, Ravanelli and Bengio, argue [9] that the optimality of these features, developed based on perceptual evidence, is not guaranteed. To addressing the above challenge, they introduced SincNet, a model employing parameterised sine functions to learn band-pass filters in its initial convolutional layer. This innovation allows SincNet to directly learn the filters' cutoff frequencies, leading to more efficient learning and faster model convergence. This approach exemplifies the shift towards learnable filters and frontends capable of directly working with raw waveforms [2]. Loweimi et al. proposed an improved SincNet by introducing more flexible and interpretable kernel-based filters, such as triangular, gammatone, and Gaussian. These filters allow for a better alignment with perceptual and statistical audio features compared to SincNet's rectangular filters [38].

In the study "CGCNN," Noé et al. [39] proposed an advancement of SincNet by incorporating complex Gabor filters and managing the resulting complex-valued signals with Complex-Valued Convolutional Neural Networks (CVCNN). This method capitalises on the superior time-frequency localisation properties of Gabor filters, tailoring them for specialised applications in audio processing.

In the progression of audio front-end development, Sainath et al. [39] marked a notable advancement by introducing a Convolutional Long Short-Term Memory Deep Neural Network (CLDNN) model trained on raw waveform data. This study demonstrated that raw waveform features, when used with a sophisticated CLDNN acoustic model, could match the performance of traditional log-mel filterbank energies. The CLDNN architecture's effectiveness, particularly its time convolution layer in reducing temporal variations and LSTM layers for temporal modelling, was a significant breakthrough. Following this trend, the study by Zeghidour et al. [40] introduced time-domain filterbanks (TD-filterbanks), a set of complex filters operating directly on the raw waveform. This approach deviated from the traditional mel-filterbank-based models, showing that TD-filterbanks, when fine-tuned within a convolutional neural network, consistently outperformed their mel-filterbank counterparts across various architectures. Furthering this trajectory, Zeghidour et al. [10] introduced "LEAF: A Learnable Frontend for Audio Classification" representing another significant step in this evolving field. LEAF introduced a fully learnable frontend that excels across various audio classification tasks, including speech, music, and environmental sounds. By deconstructing mel-filterbanks into filtering, pooling, compression, and normalisation components, LEAF offers a lightweight, adaptable architecture with far fewer parameters. It outperforms both traditional mel-filterbanks and previous learnable alternatives, demonstrating its effectiveness in multi-task settings and on large-scale benchmarks like Audioset [41]. EfficientLEAF, as proposed in a subsequent study by Schlüter and Gutenbrunner [42], addresses some of the computational inefficiencies of LEAF, particularly for long input sequences, without sacrificing accuracy on downstream tasks. This version of LEAF incorporates inhomogeneous convolution kernel sizes and strides and replaces Per-Channel Energy Normalisation (PCEN) with more parallelizable operations like logarithmic compression, temporal median subtraction, and temporal

batch normalisation. These modifications significantly improve computational performance, offering similar results to the original LEAF but at a fraction of the computational cost. However, it's important to note that neither EfficientLEAF nor LEAF consistently outperforms fixed mel filterbanks in all scenarios, suggesting that the quest for the optimal learnable audio frontend is still ongoing.

Embracing the fundamental principle of deep learning, which emphasises learning directly from raw data, a growing area of research is shifting away from using traditional features or frontends. Instead, this new direction involves directly inputting raw waveform data into neural network models, such as 1D Convolutional Neural Networks (CNNs). This approach aligns with the deep learning ethos of minimal preprocessing and reliance on the model's capability to extract relevant features autonomously. This research avenue explores various modifications and adaptations of CNN architectures [5], [43]–[45], specifically designed to effectively process and learn from 1D audio signals in their raw form [33]. Building on this approach Kim et al. [2] delve deeper into this concept by examining the SampleCNN model, an end-to-end architecture uniquely designed for processing raw waveform data. SampleCNN stands out for its use of very small filter sizes, which are particularly effective in handling various audio classification tasks, including music auto-tagging, keyword spotting, and acoustic scene tagging. The study not only demonstrates the efficacy of SampleCNN but also extends it with elements from residual and squeeze-and-excitation networks, enhancing its discriminative power and computational efficiency. This research underscores the potential of finely tuned CNN architectures in directly harnessing the nuances of raw audio data, paving the way for more sophisticated and efficient audio processing techniques.

Advancing the application of raw audio waveforms in CNN architectures, and drawing inspiration from existing research on the development of learnable filters, this study introduces CosCovNN, an innovative cosine filter-based CNN model. The design of CosCovNN was influenced by the natural compatibility of cosine functions with the periodic nature of audio signals, an idea deeply entrenched in Fourier analysis principles [46]. Motivated by this concept, we took the pioneering step of substituting traditional CNN kernels with learnable cosine filters. This key decision aligns perfectly with the inherent properties of audio waveforms and leads to a significant reduction in the model's complexity, cutting down its parameter count by approximately 77% compared to conventional CNNs. As a result, CosCovNN not only simplifies the architecture but also enhances performance, surpassing typical CNN architectures in various datasets.

Although the CosCovNN model initially outperformed standard CNNs, it encountered limitations when compared to more advanced models in contemporary research. To address this, we further improve CosCovNN by integrating Vector Quantisation and Memory modules, significantly enhancing its performance. This strategic improvement, combined with the inherent efficiency of cosine filters in processing audio signals, elevated CosCovNN to a new level of effectiveness. As a result, CosCovNN achieved state-of-the-art results, establishing it as a highly capable model for complex audio classification tasks.

III. PROPOSED RESEARCH METHOD

A. BACKGROUND KNOWLEDGE

This section presents a detailed overview of the Cosine Convolutional Neural Network architecture and its integration into the Vector Quantised Cosine Convolutional Neural Network with Memory (VQCCM) model. The VQCCM model is constructed using the CosCovNN layer as a fundamental building block.

1) Convolution in Convolutional Neural Network

1D convolutional neural networks usually consist of convolutional layers, maxpool, and fully connected layers. For any particular layer, if the input is $x[n]$, the convolutional operation can be defined as follows,

$$o[n] = x[n] * h[n] = \sum_{l=0}^{L-1} x[l] \cdot h[n-l] \tag{1}$$

where, $h[n]$ is the filter with length L and $o[n]$ is the output of that layer. During training, the aim is to learn the L parameters of the filters. Each layer of the convolutional neural network is comprised of multiple filters. We learn these filters through back-propagation from the training data during the training.

2) Vector Quantisation

The idea behind vector quantisation (VQ) is to represent n set of vectors, $V \in \{v_1, v_2, \dots v_n\}$ by a finite set of m representative vectors from a codebook, $C \in \{c_1, c_2, \dots c_m\}$. Here, each vector v_i and c_j has an equal dimension of D where $i \in \{1, 2, 3 \dots n\}$ and $j \in \{1, 2, 3 \dots m\}$ [47]. The goal of VQ is to find the closest representative vector of v_i in C and represent v_i as c_j through the mapping function G, which can be formulated as follows,

$$G(v_i) = argmin_j ||v_i - c_j||_2 \tag{2}$$

where $||v_i - c_j||_2$ represents the squared Euclidean distance between the input vector point v_i and the representative vector c_j.

VQ representation can be a very powerful layer in neural networks; however, the challenge lies in the computation of the gradient for $argmin_j ||v_i - c_j||_2$. In the VQ-VAE paper [47], authors have addressed this issue by using the gradient of $\nabla_{c_j} L$ to update the vector v_i, where L is the loss of any neural network. In this paper, the authors have used the VQ layer in their Autoencoder Network to learn the discrete representation c_j for $v_i = E(x_i)$, where E is the Encoder, and x_i is the input data. Decoder, D takes the VQ representation c_j to reconstruct x_i. Here, the reconstruction objective is $\log D(\hat{x_i} \approx x_i | c_j)$. As the dimension of c_j is equal to v_i, the gradient calculated for the c_j can be used to

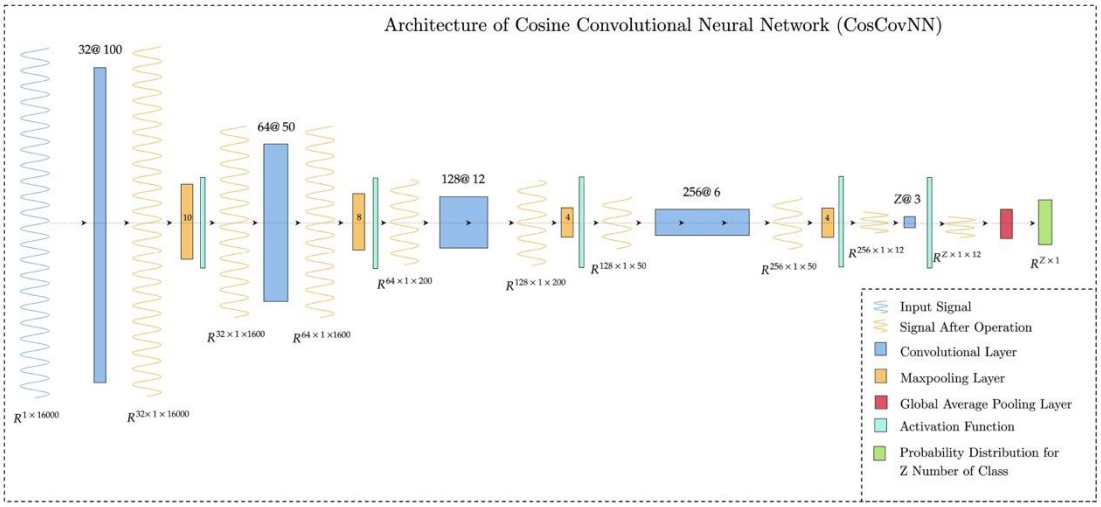

FIGURE 1. The present figure depicts an exposition of the intricate architecture of the CosCovNN model, which is designed to process a one-second audio signal with a sampling rate of 16KHz. The convolutional layer is illustrated with the number of filters denoted by the symbol @ on the left-hand side, while the filter size is indicated on the right-hand side.

update the weights of E. This way, the Autoencoder is trained end-to-end with the back-propagation algorithm. Here, the authors have the following loss function to train the VQ-VAE,

$$L = \log D(\hat{x_i} \approx x_i | c_j) + \|\text{sg}[v_i] - c_j\|_2^2 + \beta\|v_i - \text{sg}[c_j]\|_2^2, \tag{3}$$

Where, stop gradient, sg stops the flow of gradient during the back-propagation through a particular layer in a neural network and β is the hyperparameter. Here, $\beta\|v_i - \text{sg}[c_j]\|_2^2$ part forces the Encoder, E to learn v_i close to c_j and $\|\text{sg}[v_i] - c_j\|_2^2$ part makes sure that c_j does not deviate much from v_i.

B. ARCHITECTURE OF THE COSCOVNN

The 1D Convolutional Neural Network (CNN) and 1D Cosine Convolutional Neural Network (CosCovNN) differ primarily in the filter of their convolutional layers. A CNN requires learning L parameters for each particular filter of size L, whereas a CosCovNN only requires learning two parameters for a filter of the same size L. Thus, for any given filter, the CosCovNN effectively reduces the number of parameters by $L - 2$, where $L > 2$, relative to the CNN architecture. A visual comparison of the convolutional layers of the two architectures is presented in figure 2. Important components of the architecture are discussed as follows,

1) Convolutional Layer

For the convolutional layer, we generate the filter from a periodic cosine function. A cosine function can be represented as follows,

$$y[n] = A\cos(\frac{2\pi}{\lambda}n) \tag{4}$$

where, A is the Amplitude, λ is the wave length and n is the step. As, 2π is a constant, let $\frac{2\pi}{\lambda} = \theta_2$ and $A = \theta_1$. Therefore, we can represent the equation as follows,

$$g[n, \theta_1, \theta_2] = \theta_1 \cos(\theta_2 n) \tag{5}$$

Here, for any particular convolutional layer of the CosCovNet, if the input is $x[n]$, the convolutional operation can be defined as follows,

$$o[n] = x[n] * g[n, \theta_1, \theta_2] \tag{6}$$

Here, $o[n]$ is the output, θ_1, θ_2 are the learn-able parameters of the filter.

2) Pooling Layer

We have used 1D max-pooling layer for down-sampling between layers. This layer selects the most salient features within a window of size k, thereby enhancing the significance of the features obtained from the convolutional layer. [48]

3) Activation Layer

The activation function employed in CosCovNN is a crucial component of the network's architecture. As the values of the filters in CosCovNN are periodic, ranging from -1 to 1, it is imperative to maintain this range throughout the output of each layer of the network. To ensure consistency in the range of output values, we utilise the $tanh$ activation function.

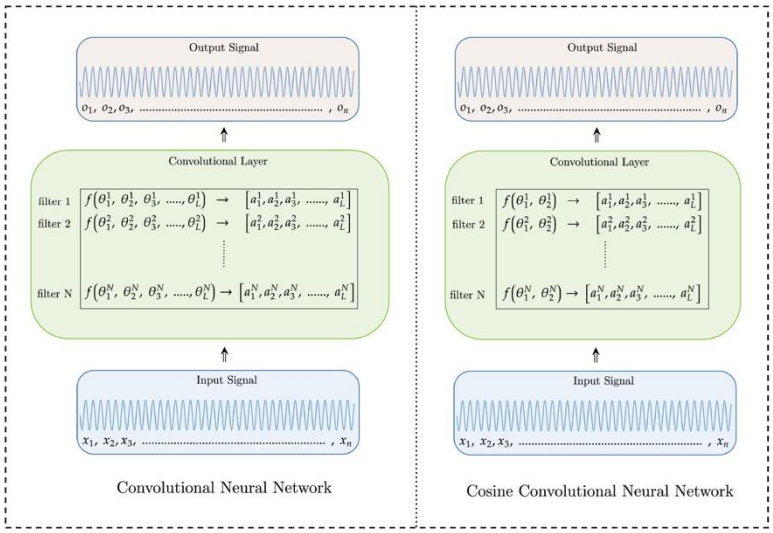

FIGURE 2. The figure illustrates a comparative analysis between the convolutional layer of a standard convolutional neural network (CNN) and that of the Cosine Convolutional Neural Network (CosCovNN). The symbols f, x, o, θ, and a denote the filter, input, output, learnable parameters, and filter values, respectively, for both models. Notably, the CosCovNN architecture has only two parameters, θ, for any given filter of size L, whereas the CNN model requires L parameters for each filter of size L.

4) Classification Layer

In classification tasks involving Z classes, the conventional approach is to employ a fully connected layer of size Z at the end of the network. However, this results in a substantial increase in the number of model parameters. To address this issue, we propose utilising Z Cosine Convolutional Layers with global average pooling [49] in the classification layer, which enables us to significantly reduce the parameter count.

5) Dropout

To enhance the resilience of the network and prevent overfitting, we incorporate 1D spatial dropout [50]. Unlike conventional dropout methods that randomly discard individual elements, spatial dropout removes entire 1D feature maps, thereby enabling the network to learn more robust and generalised features.

C. ARCHITECTURE OF THE VQCCM

The VQCCM model is an extension of the CosCovNN that incorporates Vector Quantization (VQ) and Memory Layers to improve its performance. Figure 3 illustrates the detailed architecture of the VQCCM model. In this model, the VQ layer is used in the first layer, and every layer has a memory writer and reader. During training, we learn the memory layer, as well as the reader and writer.

In the first layer, only a memory writer is present, which takes the feature from the preceding layer and writes important information to carry it to the next layer. The memory readers read the information from memory and merge it with the feature. The memory writer writes information in the memory from each layer on top of the memory obtained from the preceding layer. This allows the important information to pass from direct audio input to the output layer and in between.

1) Vector Quantisation Layer

In the VQ layer, an embedding matrix or codebook $E \in \mathbb{R}^{d,k}$ is used, where k represents the number of embedding vectors and d is the sequence length of the vector. It is noteworthy that d is identical to the sequence length of the incoming feature.

During the forward pass, the audio is passed through the Cosine Convolutional Neural Network (CosCovNN) and max-pooling layer, yielding a feature representation denoted as $F \in \mathbb{R}^{b,c,d}$, where b and c denote the number of batches and channels, respectively. Specifically, for each batch and channel, the Euclidean Distance between the feature, F_i, where $i \in \{1, 2, \dots b \times c\}$ and the embedding vectors E_j, where $j \in \{1, 2, \dots k\}$ from the codebook E, is computed. The closest vector $F_i^{'}$ is then selected from the codebook, replacing the original F_i feature and passing it to the next layer of the model. This operation can be expressed as follows,

$$F_i = F_i^{'} = E_k, \text{where}, k = argmin_j ||F_i - E_j||_2 \quad (7)$$

During the backward pass, the gradient of $F_i^{'}$ is copied to F_i as their shapes are equal, and the training process continues as usual. Figure 4 shows the architecture of the VQ layer.

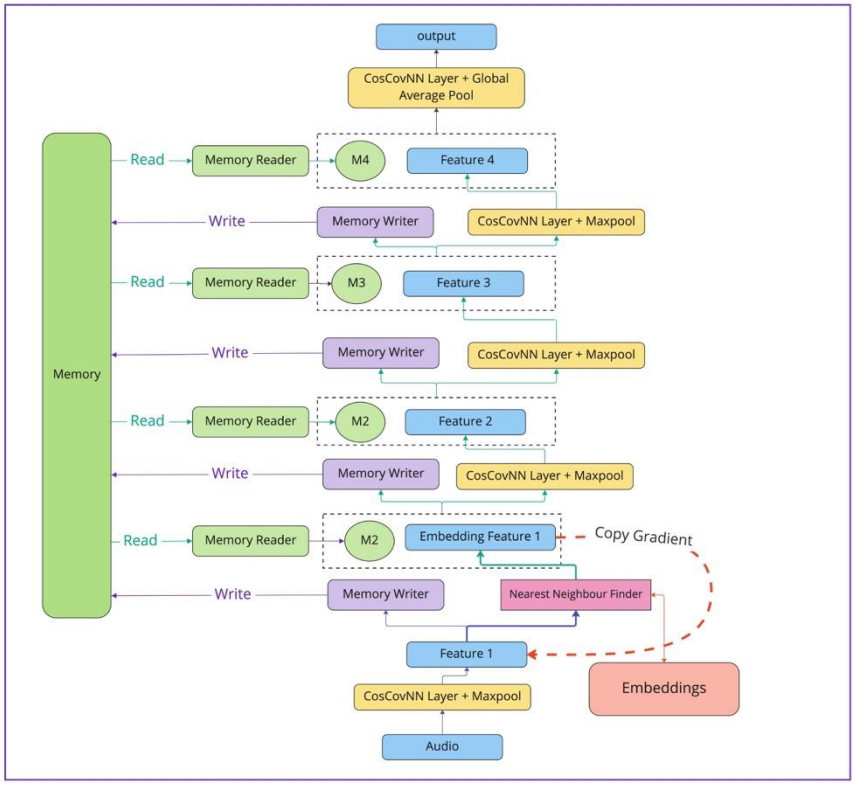

FIGURE 3. The diagram presents the structure of the VQCCM model. In this architecture, the Vector Quantisation Layer is strategically positioned right after the initial CosCovNN layer. This placement compels the model to transmit only crucial information to subsequent layers, thereby ensuring that the first layer extracts significant features due to its direct interaction with the input. On the left side of the illustration, the memory layer is depicted. This layer is connected to every other layer in the model, facilitating the transfer of information throughout the entire network, from the initial layers to the final ones.

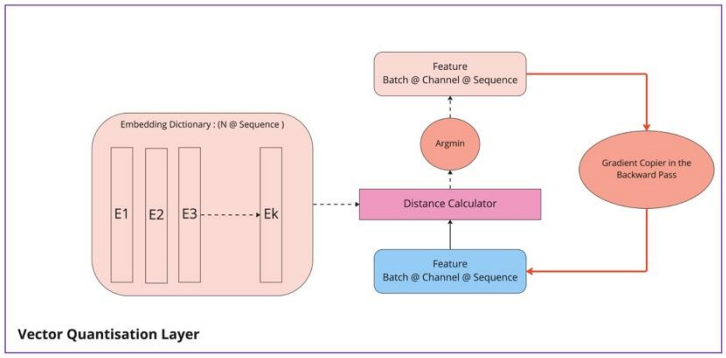

FIGURE 4. The architectural details of the VQ layer are shown in this figure. Here, the incoming feature is replaced with the nearest embedding from the codebook E. This replacement operation with $argmin$ does not have any gradient. Therefore, the gradient is copied from the replaced feature to the original feature.

2) Memory Layer

The memory layer is composed of three key components: Memory (MEM), Memory Writer (MW), and Reader (MR). The Memory vector MEM is a vector with dimensions $\mathbb{R}^{1\times M}$, where M represents the size of the Memory vector. During each layer, the Reader, MR reads the Memory vector MEM and multiplies it by the current layers feature vector f_l to get $f_l^{'}$, where l refers to the index of the layer of the VQCCM. Now, the Memory Writer, MW takes the feature vector $f_l^{'}$ and sums its intermediate representation with MEM to produce a new Memory vector, which is utilised by the subsequent layers Reading operation. By utilising these distinct components, the memory layer facilitates the efficient flow of information across the VQCCM network. Rather than initiating the memory with zero, we learn the MEM during the training and used this learned memory during the test time as the start memory. Figure 5 portrays the architecture of the Memory reader and writer blocks.

a: Memory Reader

The MR takes the memory, $MEM \in \mathbb{R}^{B,M}$ repeated over the batch of size B. The initial read operation is conducted by a Feed Forward Network, F_R. The output size of F is equal to the sequence length, S of the current layer feature $f_l \in \mathbb{R}^{B,S,C}$, where C is the number of channels. Then the output is passed through C number of CosCovNN layers (Cosine Convolutional Layer, CCL) to get the memory, $MEM \in \mathbb{R}^{B,S,C}$. Now, the MEM is multiplied with the feature f_l to get the feature $f_l^{'}$ to pass through the next layer and the Memory Writer. After both F_R and CCL layers, we use the activation function $tanh$. The whole read operation can be summarised as follows,

$$f_l^{'} = f_l \odot tanh(CCL(tanh(F_R(MEM)))) \tag{8}$$

b: Memory Writer

The Memory Writer, MW takes the feature $f_l^{'}$ and passes it through the CCL layer. Then the output is passed through Global Average Pooling, GAP to remove the dimension C from $f_l^{'}$. Now, this is passed through the Feed Forward Network F_W to get the intermediate feature of size (B, M). Finally, to write and create a new memory, the intermediate feature is added with the memory, MEM. This MEM is used in the subsequent layers read operation. Similar to MR, we use $tanh$ activation after each layer. The whole write operation can be expressed as follows,

$$MEM = MEM + tanh(F_W(GAP(tanh(CCL(f_l^{'}))))) \tag{9}$$

3) Training Objective

As both of the networks are evaluated on classification tasks, we used cross-entropy loss during the training. However, for VQCCM, we have an extra loss for the VQ layer. The total loss, L is computed as follows,

$$L = -\sum_{i=1}^{Z} y_i \log(\hat{y_i}) + \|\text{sg}[F_i] - E_j\|_2^2 + \beta\|F_i\text{sg}[E_j]\|_2^2 \tag{10}$$

Where Z is the number of classes in the classification task.

IV. DATASET

CosCovNN and VQCCM are evaluated on five datasets from both speech and non-speech audio domains. The datasets used in this study are as follows:

A. SPEECH COMMAND CLASSIFICATION

Speech Command Dataset is consisted of 105,829 utterances of length one second and there are 35 words from 2,618 speakers [51]. An audio digit classification dataset named S09, is created from this speech command dataset where it consists of utterances for different digit categories from zero to nine. Specifically, we have used this dataset for expensive experiments.

B. SPEECH EMOTION CLASSIFICATION

In our study, we utilised the IEMOCAP dataset for emotion classification. This dataset comprises a total of 12 hours of audio data, consisting of five sessions featuring two distinct speakers (one male and one female) for each session. To ensure consistency with prior research, we focused on four primary emotional states - namely angry, neutral, sad, and happy (with the excitement category consolidated with happy) [52].

C. SPEAKER IDENTIFICATION

We employed the VoxCeleb dataset [53] for the task of speaker identification/classification. This dataset consists of over 100,000 utterances (1000 hours of audio recordings) from 1251 speakers.

D. ACOUSTIC SCENES CLASSIFICATION

We have chosen the TUT Urban Acoustic Scenes 2018 dataset for our acoustic scenes classification task. The dataset comprises 24 hours of audio, which is divided into 8640 segments of 10 seconds each. The audio belongs to ten different classes, including 'Airport', 'Shopping mall', 'Metro station', 'Pedestrian street', 'Street with medium level of traffic', 'Travelling by a tram', 'Travelling by a bus', 'Travelling by an underground metro', and 'Urban park'. [54]

E. MUSICAL INSTRUMENT CLASSIFICATION

We utilised The Nsynth audio dataset to evaluate our models for the musical instrument classification task. This dataset comprises of 305,979 musical notes with a duration of four seconds, representing ten different instruments such as 'brass', 'flute', 'keyboard', 'guitar', 'mallet', 'organ', 'reed', 'string', and 'synth lead', along with one vocal class [55].

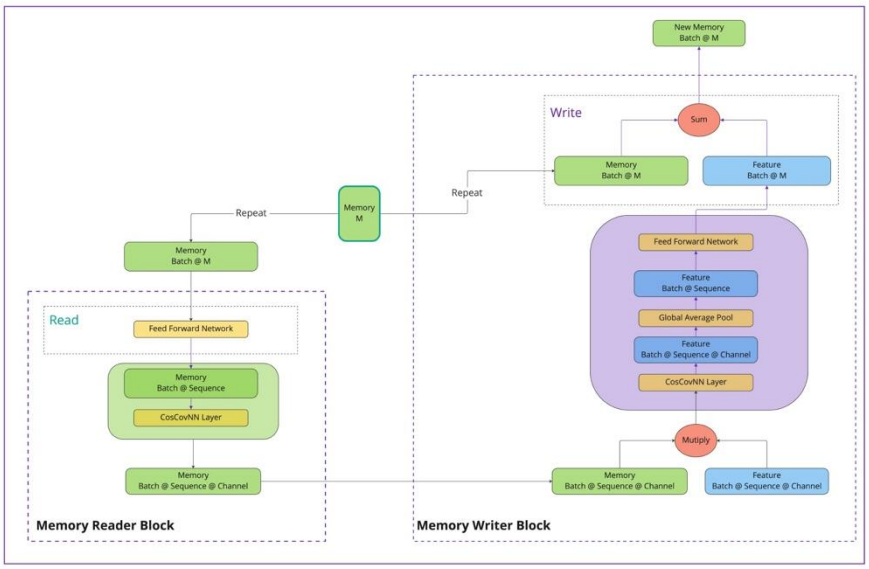

FIGURE 5. The figure provides a detailed view of the Memory Reader and Writer components within the VQCCM architecture. The Memory, labeled as M, is situated centrally between the two blocks. The Reader block accesses the memory, integrating it through a series of feed forward networks and a CosCovNN layer. This processed information is then relayed to the Writer block, where it merges the existing memory with the current layer's features, subsequently generating a new memory state. This updated memory state supersedes the previous one and is passed on to the subsequent layer for further processing

V. EXPERIMENTAL SETUP, RESULTS AND DISCUSSION

We performed experiments to assess the effectiveness of both CosCovNN and VQCCM. Our evaluation of CosCovNN involved identifying an appropriate baseline architecture to compare its performance with similar CNN architectures. Additionally, we established an experimental setup to analyse the performance of VQCCM relative to state-of-the-art literature.

A. MODEL ARCHITECTURE SEARCH FOR COSCOVNN

To evaluate the performance of cosine convolutional filters, we need to first find a benchmark architecture for the model. This will allow us to evaluate its performance and computational complexity with similar CNN architecture.

1) Experimental Setup

Like any typical CNN model, finding a suitable architecture for CosCovNN is the most challenging part. It is very common to tune the architecture (eg. change the size of kernels, number of layers, and number of filters) of any CNN model according to the datasets. However, tuning the proposed CosCovNN for different datasets is out of the scope of this research work. Therefore, we want to search for an optimal architecture for any single dataset, then measure its performance on different datasets by changing only the number of filters and layers. To find this architecture, we experiment with the S09 dataset.

First, we fix a backbone network and then change different settings to find the optimal architecture. For the backbone network (BBN), we choose five layers for the CosCovNN with filter size 12 and incremental number of filters as 32, 64, 128, 256 and 10 (number of class for S09 is 10) respectively from layer 1 to 5. For the activation function we use tanh function and maxpool at each layer of size 2 where Dropout is 50 percent.

To find suitable filter size for each layer, we follow the following strategy,

- Step 1: choose layer 1 from the BBN.
- Step 2: get classification accuracy changing the size of the filter F, to 3, 6, 12, 25, 50, 100, 200, 300 (while rest of the layers are unchanged in the BBN).
- Step 3: choose the filter F_{best}, with highest accuracy and replace the filter F of the layer with F_{best} (while rest of the layers are unchanged in the BBN).
- Step 4: choose next layer from the BBN.
- Step 5: Go back to step 2 if this not the last layer.

After we find the optimal filter size for each of the layers, we follow a similar strategy for the max-pooling layer. We explored 2, 4, 6, 8, 10, 20 window sizes for each layer. Here, every experiment is conducted five times and the maximum accuracy is recorded for comparison. This approach allowed us to identify the maximum accuracy achievable for any given setting.

TABLE 1. Classification Accuracy of CosCovNN on the S09 dataset for different filter size

Filter Size	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
3	78.64	79.54	83.67	84.16	**85.22**
6	81.34	81.93	84.16	**84.24**	84.52
12	81.89	82.12	**84.20**	84.20	84.24
25	82.00	83.54	84.12	83.50	84.05
50	82.09	**84.20**	83.10	82.09	83.77
100	**82.12**	83.97	81.97	81.77	83.65
200	82.05	83.14	81.85	81.62	83.58
300	82.01	81.81	81.30	81.58	83.54

TABLE 2. Classification Accuracy of CosCovNN on the S09 dataset for different window size of the Maxpool

Pool Size	Layer 1	Layer 2	Layer 3	Layer 4
2	85.22	93.81	95.37	95.53
4	89.53	94.43	**95.53**	**96.32**
8	92.71	**95.37**	95.22	96.00
10	**93.81**	92.98	94.86	95.69
20	93.57	91.77	94.43	95.34

2) Results

The results presented in tables 1, we observed that a gradual reduction in filter size from layer 1 to 5 yielded better performance. This is likely due to a decrease in feature size resulting from maxpooling at each layer, which allows for better capturing of key features with smaller filter sizes.

For maxpooling window sizes, we identified optimal values of 10, 8, 4, and 4 for layers 1 to 4, respectively. Our analysis indicates that larger pooling sizes can lead to significantly improved accuracy. However, balancing pooling sizes across different layers is essential to avoid performance degradation. These results provide valuable insights into optimising the architecture of the CosCovNN model for improved accuracy.

B. COMPARISON BETWEEN COSCOVNN AND CNN

1) Experimental Setup

We have evaluated the CosCovNN on the five datasets, where all of these datasets comes with test data except IEMOCAP data. For IEMOCAP, we have calculated the accuracy based on the five fold cross validation (each fold is a session). To get a fare comparison of the performance of CosCovNN, we have used CNN with similar architecture and assessed the number of parameters for both models. Moreover, we have also compared the results with related literature Time-Domain Filterbanks (TD-filterbank) [40], SincNet [9] and LEAF [10]. Our objective is to surpass the accuracy of the CNN with our CosCovNN model while achieving accuracy levels close to those of the related literature. Additionally, visualise the filters for comparison.

To accommodate audio signals of varying lengths, an additional layer has been incorporated at the beginning of the architecture. This layer serves to adjust the feature size to match that of a 16KHz sample rate audio signal with a duration of one second. For instance, to process audio signals with a duration of 10 seconds, a layer with a pooling size of 10 has been added. We collected the accuracy of the TD-filterbank and SincNet from the research work of Neil et al. [40], and to keep the experiment fair, we have followed the exact experimental setup from this research work. In this work, IEMOCAP and S09 dataset was not used; therefore, we have trained time-domain filterbanks, SincNet and LEAF on these datasets to collect the accuracy.

2) Results

The results of our experiments are presented in Table 3. It is observed that CosCovNN outperforms CNN for all the tasks. Moreover, CosCovNN performs better than TD-fbanks and SincNet for all the tasks except for Acoustic Scenes and Speaker Id classification, respectively. For Acoustic Scenes classification, TD-fbanks outperforms CosCovNN. Since SincNet is explicitly designed for speaker classification, it is reasonable that it achieves better classification accuracy than CosCovNN in this regard. However, CosCovNN could not outperform LEAF in any of the tasks. This suggests that CosCovNN needs some architectural changes to surpass the best-performing model, but it can still achieve close to SOTA results.

Now, we can calculate the number of parameters for the CosCovNN as $(1 \times 32 \times 2) + (32 \times 64 \times 2) + (64 \times 128 \times 2) + (128 \times 256 \times 2) + (256 \times 10 \times 2) = 91,200$ and for CNN as $(1 \times 32 \times 100) + (32 \times 64 \times 50) + (64 \times 128 \times 12) + (128 \times 256 \times 6) + (256 \times 10 \times 3) = 4,08,192$. Notably, the CosCovNN architecture has 77.66% fewer parameters than the CNN architecture, yet it outperforms the CNN. These results suggest that the cosine filters used in CosCovNN are both effective and more computationally efficient than the CNN filters in the case of audio data modelling. The fundamental difference between CosCovNN and CNN filters is illustrated in figure 6. Cosine filters are periodic, capturing critical frequency information in audio signals and are less impacted by noise. On the other hand, CNN filters try to capture the shape of the audio signal. Therefore filters are more susceptible to noise in the audio signal and less periodic.

C. COMPARISON OF VQCCM WITH LITERATURE

1) Experimental Setup

Based on our previous experiments, we found that the CosCovNN architecture with cosine filters is more efficient than raw CNN filters. However, it is still being determined whether this approach can be used to develop a robust model that can achieve or beat SOTA performance in the literature. To address this question, we augmented the CosCovNN architecture with Memory and VQ layers to create VQCCM. We trained VQCCM with similar datasets and aimed to surpass the performance of LEAF, the best-performing model. We aim to demonstrate that the CosCovNN architecture with Memory and VQ layers can be a powerful model for audio classification tasks and achieve or surpass SOTA performance.

TABLE 3. Comparison of CosCovNN and VQCCM with the literature based on the test classification accuracy for different tasks

Classification Task	TD-fbanks	SincNet	CNN	CosCovNN	LEAF	VQCCM
Speech Command	87.3 ± 0.4	89.2 ± 0.4	83.1 ± 0.5	91.5 ± 0.2	93.4 ± 0.3	**95.6 ± 0.1**
Spoken Digit	94.6 ± 0.3	95.4 ± 0.2	91.4 ± 0.2	96.3 ± 0.1	96.7 ± 0.2	**97.1 ± 0.2**
Speech Emotion	58.7 ± 1.4	59.5 ± 3.2	54.2 ± 1.2	63.1 ± 2.8	66.8 ± 1.8	**71.2 ± 1.4**
Acoustic Scenes	**99.5 ± 0.4**	96.7 ± 0.9	95.6 ± 0.7	98.3 ± 0.6	99.1 ± 0.5	99.1 ± 0.3
Musical Instrument	70.0 ± 0.6	70.3 ± 0.6	68.3 ± 0.9	71.5 ± 0.2	72.0 ± 0.6	**73.1 ± 0.1**
Speaker Id	25.3 ± 0.7	43.5 ± 0.8	17.4 ± 3.4	31.4 ± 0.9	33.1 ± 0.7	**47.7 ± 0.6**

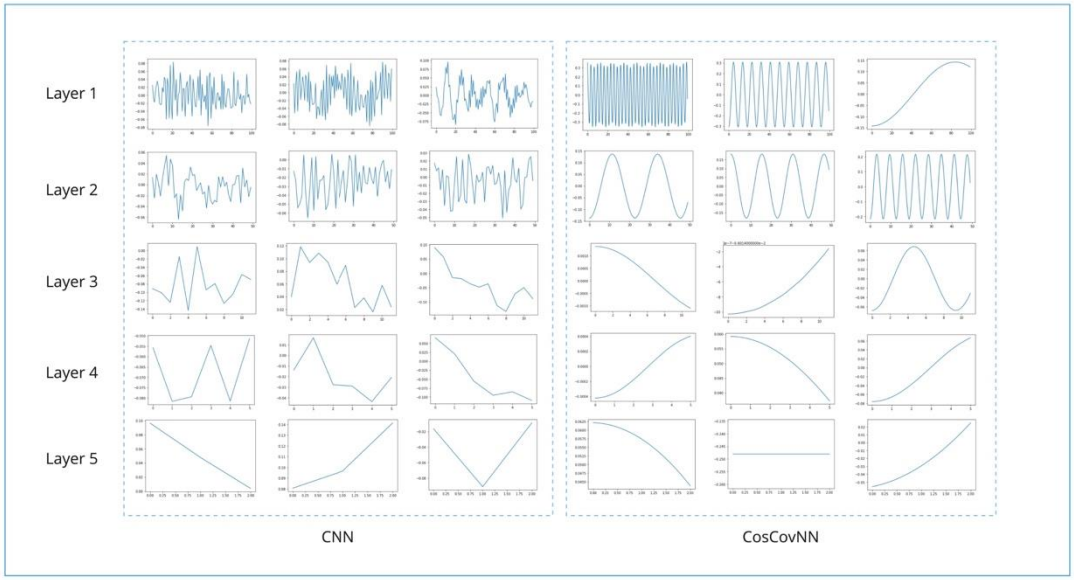

FIGURE 6. The figure presents a comparative visualization of the trained filter outputs from CosCovNN and traditional CNNs. On the left, the CNN filters appear irregular, reflecting their adaptation to the intricate patterns within the audio waveform through the training process. On the right, the CosCovNN filters exhibit a smoother contour, suggesting that during training, these filters have emphasised on the periodic characteristics or frequency aspects of the waveform.

2) Results

As shown in Table 3, VQCCM has outperformed LEAF for all tasks. However, neither LEAF nor VQCCM could exceed TD-Fbanks performance in acoustic scene classification. VQCCM and LEAF achieved similar accuracy of 99.1%, but VQCCM has a lower standard deviation than LEAF. Furthermore, as we have only tuned VQCCM with the S09 dataset, there is an opportunity for researchers to explore and fine-tune VQCCM for each problem separately. These results demonstrate that the Cosine Convolution filter can be a solid alternative to CNN filters for raw audio classification.

D. IMPACT OF MEMORY AND VQ SIZE

1) Experimental Setup

The VQCCM model is designed to enhance information propagation from the lower layers to the classifier layer by utilising its memory component, where the VQ layer is responsible for learning representations from specific embedding vectors. The number of vectors in the VQ layer and the size of the memory layer are two crucial factors that significantly influence the performance of VQCCM. To identify the appropriate memory size and VQ embedding numbers, we conducted separate training experiments by integrating the memory and VQ layer into the CosCovNN architecture. Initially, we planned to utilise the S09 dataset for this experimentation. However, as the performance of VQCCM and CosCovNN was found to be very similar on this dataset, the impact of the memory and VQ layer might be clear from the comparison. As a result, we expanded our experiments to include the IEMOCAP dataset. Here, we assessed the maximum accuracy based on five runs and plotted it on a graph to gain insights into its behaviour.

2) Results

The experimental outcomes are illustrated in Figure 7. Our results demonstrate that the model's performance can be significantly enhanced by integrating memory into it. However,

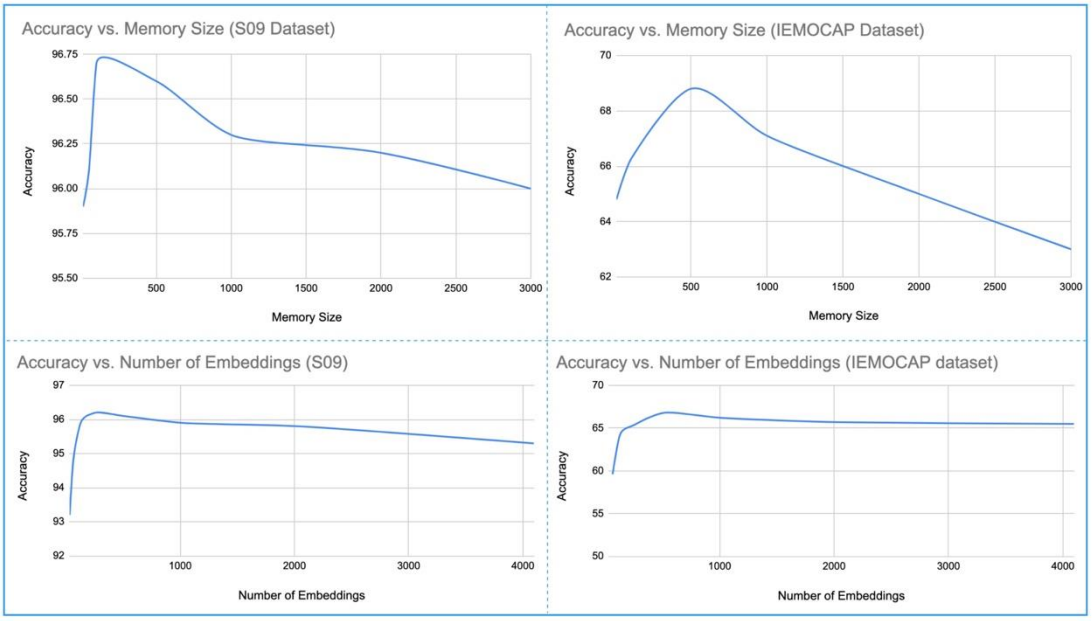

FIGURE 7. This figure shows the impact of the Memory and VQ layer of the VQCCM model for the S09 and IEMOCAP datasets.

it is imperative to note that an excessive increase in memory size can lead to overfitting, thereby causing a decline in performance. Conversely, setting the memory size too small, such as 10, may impair the performance of VQCCM by inadequately representing vital information from previous layers. In such cases, the multiplication of memory with the learned features in each layer negatively affects the VQCCM's performance. Prior to integrating either memory or VQ layers, our previous experiments on the S09 and IEMOCAP datasets revealed maximum accuracies of 96.4% and 65.9%, respectively, for CosCovNN. After the inclusion of the memory layer, we achieved a maximum accuracy of 96.7% for the S09 dataset with a memory size of 100 and a maximum accuracy of 66.8% for the IEMOCAP dataset with a memory size of 500. Similarly, we obtained accuracies of 96.2% and 67.0% for the S09 and IEMOCAP datasets, respectively, with embedding sizes of 256 and 512. We observed that using a lower number of embeddings can result in underfitting while increasing the embedding size beyond a certain threshold does not cause significant degradation in performance. Unlike memory layers, overfitting the VQ layer depends on the number of feature layers, and a higher number of embeddings does not lead to overfitting.

E. ABLATION STUDY FOR VQCCM

In order to investigate the role of the Memory and VQ layer in the VQCCM model, we conducted an ablation study. This involved adding each component separately to the CosCovNN and observing the impact on model performance. The ablation study is equivalent to removing each component from VQCCM individually. Specifically, we performed two experiments: the first involved adding the Memory layer to the CosCovNN, resulting in a model referred to as VQCCM - VQ, and the second involved adding the VQ layer to the CosCovNN, resulting in a model referred to as VQCCM - Memory. The results of these experiments are presented in Table 4.

Our findings indicate that adding the Memory layer to the CosCovNN results in improved performance and stability compared to CosCovNN. However, when only the VQ layer is added to the CosCovNN, improvements are not consistently observed across all experiments. While Memory is a valuable addition to CosCovNN, it is even more effective when combined with the VQ layer. The VQ layer enforces the use of a fixed number of vectors, making it difficult for the model to learn an effective representation. However, by adding the Memory layer, the model is forced to use its memory to pass important information that cannot be learned through the VQ layer alone. As a result, the presence of the VQ layer compels the model to utilise the Memory layer, leading to better results.

VI. CONCLUSION

In this research, we introduced the concept of using cosine filters as a replacement for conventional filters in Convolutional Neural Network models to classify audio directly

TABLE 4. Ablation study of the VQCCM (CosCovNN + Memory + VQ)

Classification Task	CosCovNN	CosCovNN + Memory	CosCovNN + VQ	CosCovNN + Memory + VQ
Speech Command	91.5 ± 0.2	94.2 ± 0.1	92.9 ± 0.5	$\mathbf{95.6 \pm 0.1}$
Spoken Digit	96.3 ± 0.1	96.5 ± 0.1	96.1 ± 0.2	$\mathbf{97.1 \pm 0.2}$
Speech Emotion	63.1 ± 2.8	68.1 ± 0.7	64.1 ± 2.9	$\mathbf{71.2 \pm 1.4}$
Acoustic Scenes	98.3 ± 0.6	98.7 ± 0.3	98.1 ± 0.4	$\mathbf{99.1 \pm 0.3}$
Musical Instrument	71.5 ± 0.2	71.9 ± 0.1	70.9 ± 0.8	$\mathbf{73.1 \pm 0.1}$
Speaker Id	31.4 ± 0.9	38.2 ± 0.4	30.6 ± 1.6	$\mathbf{47.7 \pm 0.6}$

from raw waveforms. A major benefit of cosine filters is their computational simplicity, as any particular filter requires learning only two parameters. This contrasts with traditional CNN filters, where the number of parameters varies and is typically higher. To implement this approach, we developed the CosCovNN model, which integrates cosine filters into the CNN framework. Through comparative analyses of CosCovNN and similar CNN architectures, conducted on both Speech and NonSpeech datasets, we have demonstrated that CosCovNN is an effective alternative for audio classification directly from raw waveforms. Our study details the necessary modifications for incorporating cosine filters into Convolutional Neural Network (CNN) models. This provides a clear pathway for researchers to adapt existing CNN architectures, as found in the literature, to include cosine filters. By implementing the changes we suggest, these modified CNN models could potentially benefit from a reduced number of parameters and enhanced performance. This aspect of our research opens up promising opportunities for future investigations, especially developing different CNN architecture based on cosine filters to model audio from raw waveform.

Morevoer, in this study we proposed The VQCCM model, which is an enhancement of the CosCovNN framework, incorporating a Vector Quantisation (VQ) layer and a Memory Layer. We evaluated the classification performance of VQCCM across five different datasets (Speech Command, Speech Emotion, Acoustic Scenes, Musical Intrument and Speaker Id), where we achieved state-of-the-art results and outperform benchmarks in certain instances. The integration of the VQ layer and memory module in VQCCM significantly enhances the performance of the CosCovNN model. This innovative approach paves the way for researchers to apply these combined elements in various CNN architectures found in the literature. While VQCCM has demonstrated promising results in our datasets, its potential for broader application and exploration in other domains remains an exciting prospect for future research.

REFERENCES

[1] L. Jiao and J. Zhao, "A survey on the new generation of deep learning in image processing," IEEE Access, vol. 7, pp. 172231–172263, 2019.
[2] T. Kim, J. Lee, and J. Nam, "Comparison and analysis of samplecnn architectures for audio classification," IEEE Journal of Selected Topics in Signal Processing, vol. 13, no. 2, pp. 285–297, 2019.
[3] J. Sang, S. Park, and J. Lee, "Convolutional recurrent neural networks for urban sound classification using raw waveforms," in 2018 26th European Signal Processing Conference (EUSIPCO), pp. 2444–2448, 2018.
[4] J.-w. Jung, H.-S. Heo, J.-h. Kim, H.-j. Shim, and H.-J. Yu, "Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," in Interspeech, pp. 1268–1272, 2019.
[5] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for lvcsr," in Fifteenth annual conference of the international speech communication association, Citeseer, 2014.
[6] N. Liang, W. Xu, C. Luo, and W. Kang, "Learning the front-end speech feature with raw waveform for end-to-end speaker recognition," in Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence, ICCAI '20, (New York, NY, USA), p. 317–322, Association for Computing Machinery, 2020.
[7] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2721–2725, 2017.
[8] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 421–425, 2017.
[9] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 1021–1028, IEEE, 2018.
[10] N. Zeghidour, O. Teboul, F. de Chaumont Quitry, and M. Tagliasacchi, "{LEAF}: A learnable frontend for audio classification," in International Conference on Learning Representations, 2021.
[11] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," IEEE Transactions on Computers, vol. C-23, no. 1, pp. 90–93, 1974.
[12] H. Chen, P. Zhang, and Y. Yan, "An audio scene classification framework with embedded filters and a dct-based temporal module," in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 835–839, 2019.
[13] C. Geng and L. Wang, "End-to-end speech enhancement based on discrete cosine transform," in 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), pp. 379–383, IEEE, 2020.
[14] J. O. Smith, Mathematics of the discrete Fourier transform (DFT): with audio applications. Julius Smith, 2008.
[15] A. Ramalingam and S. Krishnan, "Gaussian mixture modeling of short-time fourier transform features for audio fingerprinting," IEEE Transactions on Information Forensics and Security, vol. 1, no. 4, pp. 457–463, 2006.
[16] N. Yang, M. Usman, X. He, M. A. Jan, and L. Zhang, "Time-frequency filter bank: A simple approach for audio and music separation," IEEE Access, vol. 5, pp. 27114–27125, 2017.
[17] H. Lee, Y.-H. Wu, Y.-S. Lin, and S.-Y. Chien, "Convolutional neural network accelerator with vector quantization," in 2019 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5, 2019.
[18] S. Park, S. Kim, S. Lee, H. Bae, and S. Yoon, "Quantized memory-augmented neural networks," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, Apr. 2018.
[19] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in Proceedings

of The 33rd International Conference on Machine Learning (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of Proceedings of Machine Learning Research, (New York, New York, USA), pp. 1842–1850, PMLR, 20–22 Jun 2016.

[20] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," arXiv preprint arXiv:1410.5401, 2014.

[21] G. Guo and S. Z. Li, "Content-based audio classification and retrieval by support vector machines," IEEE transactions on Neural Networks, vol. 14, no. 1, pp. 209–215, 2003.

[22] L. Chen, S. Gunduz, and M. T. Ozsu, "Mixed type audio classification with support vector machine," in 2006 IEEE international conference on multimedia and expo, pp. 781–784, IEEE, 2006.

[23] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," IEEE Transactions on speech and audio processing, vol. 10, no. 7, pp. 504–516, 2002.

[24] V. Mitra and C.-J. Wang, "Content based audio classification: a neural network approach," Soft Computing, vol. 12, pp. 639–646, 2008.

[25] C. Freeman, R. Dony, and S. Areibi, "Audio environment classication for hearing aids using artificial neural networks with windowed input," in 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing, pp. 183–188, IEEE, 2007.

[26] X. Shao, C. Xu, and M. S. Kankanhalli, "Applying neural network on the content-based audio classification," in Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint, vol. 3, pp. 1821–1825, IEEE, 2003.

[27] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey, and P. Tiwari, "Sound classification using convolutional neural network and tensor deep stacking network," IEEE Access, vol. 7, pp. 7717–7727, 2019.

[28] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al., "Cnn architectures for large-scale audio classification," in 2017 ieee international conference on acoustics, speech and signal processing (icassp), pp. 131–135, IEEE, 2017.

[29] H. Yang and W.-Q. Zhang, "Music genre classification using duplicated convolutional layers in neural networks.," in Interspeech, pp. 3382–3386, 2019.

[30] D. N. Makropoulos, A. Tsiami, A. Prospathopoulos, D. Kassis, A. Frantzis, E. Skarsoulis, G. Piperakis, and P. Maragos, "Convolutional recurrent neural networks for the classification of cetacean bioacoustic patterns," in ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5, IEEE, 2023.

[31] T. Makino, H. Liao, Y. Assael, B. Shillingford, B. Garcia, O. Braga, and O. Siohan, "Recurrent neural network transducer for audio-visual speech recognition," in 2019 IEEE automatic speech recognition and understanding workshop (ASRU), pp. 905–912, IEEE, 2019.

[32] Z. Zhang, S. Xu, S. Zhang, T. Qiao, and S. Cao, "Attention based convolutional recurrent neural network for environmental sound classification," Neurocomputing, vol. 453, pp. 896–903, 2021.

[33] K. Zaman, M. Sah, C. Direkoglu, and M. Unoki, "A survey of audio classification using deep learning," IEEE Access, 2023.

[34] T. Arias-Vergara, P. Klumpp, J. C. Vasquez-Correa, E. Nöth, J. R. Orozco-Arroyave, and M. Schuster, "Multi-channel spectrograms for speech processing applications using deep learning methods," Pattern Analysis and Applications, vol. 24, pp. 423–431, 2021.

[35] P. C. Vakkantula, Speech Mode Classification using the Fusion of CNNs and LSTM Networks. West Virginia University, 2020.

[36] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 4052–4056, IEEE, 2014.

[37] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5115–5119, IEEE, 2016.

[38] E. Loweimi, P. Bell, and S. Renals, "On learning interpretable cnns with parametric modulated kernel-based filters.," in Interspeech, pp. 3480–3484, 2019.

[39] P.-G. Noé, T. Parcollet, and M. Morchid, "Cgcnn: Complex gabor convolutional neural network on raw speech," in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7724–7728, IEEE, 2020.

[40] N. Zeghidour, N. Usunier, I. Kokkinos, T. Schaiz, G. Synnaeve, and E. Dupoux, "Learning filterbanks from raw speech for phone recognition," in 2018 IEEE international conference on acoustics, speech and signal Processing (ICASSP), pp. 5509–5513, IEEE, 2018.

[41] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 776–780, IEEE, 2017.

[42] J. Schlüter and G. Gutenbrunner, "Efficientleaf: A faster learnable audio frontend of questionable use," in 2022 30th European Signal Processing Conference (EUSIPCO), pp. 205–208, IEEE, 2022.

[43] D. Palaz, R. Collobert, et al., "Analysis of cnn-based speech recognition system using raw speech as input," tech. rep., Idiap, 2015.

[44] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 4624–4628, IEEE, 2015.

[45] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and A. Senior, "Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms," in 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 30–36, IEEE, 2015.

[46] A. O. Salau, I. Oluwafemi, K. F. Faleye, and S. Jain, "Audio compression using a modified discrete cosine transform with temporal auditory masking," in 2019 international conference on signal processing and communication (ICSC), pp. 135–142, IEEE, 2019.

[47] A. Van Den Oord, O. Vinyals, et al., "Neural discrete representation learning," Advances in neural information processing systems, vol. 30, 2017.

[48] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," Mechanical systems and signal processing, vol. 151, p. 107398, 2021.

[49] M. Lin, Q. Chen, and S. Yan, "Network in network," arXiv preprint arXiv:1312.4400, 2013.

[50] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 648–656, 2015.

[51] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," CoRR, vol. abs/1804.03209, 2018.

[52] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," Language resources and evaluation, vol. 42, no. 4, pp. 335–359, 2008.

[53] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," Computer Speech and Language, vol. 60, p. 101027, 2020.

[54] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in Workshop on Detection and Classification of Acoustic Scenes and Events, 2018.

[55] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in International Conference on Machine Learning, pp. 1068–1077, PMLR, 2017.

KAZI NAZMUL HAQUE is currently pursuing his PhD at the University of Southern Queensland in Australia. Alongside his academic endeavors, he holds the position of Senior Machine Learning Engineer at Splash Music in Australia, where he specializes in the development of generative models for music. With over eight years of professional experience in machine learning, Kazi's expertise is grounded in applying machine learning techniques to address a variety of real-world challenges. His present research is primarily centered on unsupervised representation learning for audio data. Prior to embarking on his doctoral studies, Kazi earned a Master's degree in Information Technology from Jahangirnagar University in Bangladesh.

RAJIB RANA is an experimental computer scientist, Advance Queensland Research Fellow and a Professor in the University of Southern Queensland. He is also the Director of the IoT Health research program at the University of Southern Queensland. He is the recipient of the prestigious Young Tall Poppy QLD Award 2018 as one of Queensland's most outstanding scientists for achievements in the area of scientific research and communication. Rana's research work aims to capitalise on advancements in technology along with sophisticated information and data processing to better understand disease progression in chronic health conditions and develop predictive algorithms for chronic diseases, such as mental illness and cancer. His current research focus is on Unsupervised Representation Learning. He received his B. Sc. degree in Computer Science and Engineering from Khulna University, Bangladesh with the Prime Minister and President's Gold medal for outstanding achievements and a Ph. D. in Computer Science and Engineering from the University of New South Wales, Sydney, Australia in 2011. He received his postdoctoral training at Autonomous Systems Laboratory, CSIRO before joining the University of Southern Queensland as Faculty in 2015.

BJÖRN W. SCHULLER (M'05-SM'15-F'18) received his diploma in 1999, his doctoral degree for his study on Automatic Speech and Emotion Recognition in 2006, and his habilitation and Adjunct Teaching Professorship in the subject area of Signal Processing and Machine Intelligence in 2012, all in electrical engineering and information technology from TUM in Munich/Germany. He is Professor of Artificial Intelligence in the Department of Computing at the Imperial College London/UK, where he heads GLAM — the Group on Language, Audio & Music, Full Professor and head of the Chair of Embedded Intelligence for Health Care and Wellbeing at the University of Augsburg/Germany, and founding CEO/CSO of audEERING. He was previously full professor and head of the Chair of Complex and Intelligent Systems at the University of Passau/Germany. Professor Schuller is Fellow of the IEEE, Golden Core Member of the IEEE Computer Society, Fellow of the ISCA, Senior Member of the ACM, President-emeritus of the Association for the Advancement of Affective Computing (AAAC), and was elected member of the IEEE Speech and Language Processing Technical Committee. He (co-)authored 5 books and more than 900 publications in peer-reviewed books, journals, and conference proceedings leading to more than overall 30 000 citations (h-index = 82). Schuller is Field Chief Editor of Frontiers in Digital Health, former Editor in Chief of the IEEE Transactions on Affective Computing, and was general chair of ACII 2019, co-Program Chair of Interspeech 2019 and ICMI 2019, repeated Area Chair of ICASSP, next to a multitude of further Associate and Guest Editor roles and functions in Technical and Organisational Committees.

5.3. Links and implications

In this final chapter, I introduced two novel models, Cosine Convolutional Neural Network (CosCovNN) and Vector Quantised Cosine Convolutional Neural Network with Memory (VQCCM). These models are designed to classify audio data directly from its waveform, eliminating the need for manual feature engineering. Results, based on diverse datasets, confirm that the cosine filter can serve as a viable alternative to the traditional CNN filter for raw audio waveform classification. With the completion of this chapter, the third research objective is achieved.

In the concluding section, I will summarize the key findings and contributions of this thesis. Additionally, I will outline potential future directions and areas for further research in the domain of audio processing and machine learning.

CHAPTER 6: DISCUSSION AND CONCLUSION

This study makes a significant contribution to the domain of unsupervised disentangled representation learning, a fundamental area in machine learning. The primary objective of this research is to learn disentangled representations from unlabelled audio data. These learned representations hold the potential to enhance the performance of machine learning models in audio-related tasks, particularly in scenarios where labelled data is limited. To attain this goal, I have devised innovative models tailored to operate with audio spectrogram representations. It is worth noting that manually crafted features, such as spectrograms, have been criticised for potentially limiting the performance of machine learning models. Hence, this research also has a secondary aim to address this limitation by creating models capable of directly processing raw audio waveforms. To accomplish both the primary and secondary objectives of this research, three core objectives have been formulated. Next, I will conclude my findings based on these objectives.

6.1. Guided Representation Learning from unlabelled audio data

The first core objective of this research focused on developing models capable of learning task-specific representations from unlabelled audio data while leveraging guidance from a small amount of labelled data. This approach aimed to enhance the performance of machine learning models on related tasks with limited labelled data. The Guided Generative Adversarial Neural Network (GGAN) was introduced as a solution to this objective. Even though GGAN was primarily designed for representation learning, it has also made a significant contribution to the field of audio generation. GGAN has introduced an innovative approach to guiding a GAN model to generate high-quality conditional audio based on a small set of labelled samples. The effectiveness of GGAN has been demonstrated through experiments conducted on both speech and non-speech datasets.

One of the core contributions of GGAN lies in the concept of partitioning the latent space of the GAN model based on the supervision given from a limited set of labelled data. This novel approach has implications beyond audio processing and represents a valuable contribution to machine learning theory. Researchers across various domains can leverage this idea to develop other GAN-based models capable

of learning guided representations from unlabelled datasets. This contribution transcends the audio field and can be applied to a wide range of research areas where GAN-based models play a role.

6.2. Guided and Generalised Representation Learning from unlabelled audio data

Indeed, GGAN has demonstrated its ability to disentangle specific characteristics of the data distribution in its latent space, guided by the provided supervision. This capability enhances its performance in the task at hand, particularly when that task is closely related to the supervised signal. However, this specialised partitioning of the latent space comes at a cost – it limits the capacity of GGAN to capture other variational factors of the data that are unrelated to the provided supervision signal. While GGAN's design was intentional and serves its primary purpose effectively, it also means that it may miss out on the opportunity to be used in a more generalised manner across a broader range of tasks.

Recognising this untapped potential, I introduced objective 2 as a means to complement the primary aim of this research. Nevertheless, it posed a substantial challenge to design a model based on GGAN architecture capable of simultaneously achieving both guided and generalised representation learning. In the initial stages of this research, the incorporation of both these objectives into the GGAN design proved counterproductive, resulting in a model that was ineffective for either task. To address this challenge, I transitioned to an autoencoder-based model and developed the Guided Adversarial Autoencoder (GAAE). In the GAAE architecture, the encoder learns to project high-dimensional data distributions into two distinct latent or representation spaces. Within these latent spaces, one representation captures the specific characteristics of the data guided by the provided signal, while the other focuses on capturing the generalised characteristics of the data distribution. Subsequently, the decoder can effectively reconstruct the input data distribution from these dual representations. An interesting aspect of the GAAE model is that by manipulating the values within these two representation spaces, the decoder can be repurposed as a high-fidelity audio generation model. This dual functionality highlights the versatility and potential of GAAE in both representation learning and audio generation tasks.

I conducted extensive evaluations of GAAE using three distinct datasets, and the results demonstrated its superior performance compared to existing models in the literature. GAAE exhibited the capability to generate high-fidelity conditional audio samples and simultaneously learn both guided and generalised representations, all while utilising supervision from just 1% of labelled data. Furthermore, I highlighted an additional practical application of GAAE, where it can serve as a valuable data augmentation tool by leveraging the high-quality samples it generates. Beyond its specific applications in audio processing, GAAE contributes to the foundational autoencoder theory in the field of machine learning, opening up possibilities for its utilisation in various domains where autoencoders are relevant. This versatility underscores the broader significance and potential of GAAE beyond the scope of this research.

6.3. Direct modelling of audio from raw waveform

My proposed models, GGAN and GAAE, undeniably contribute to the advancement of representation learning and audio generation in the field of machine learning. During the design of these models, I drew inspiration from the success of using spectrograms as input features for deep learning models in the existing literature. Both GGAN and GAAE are tailored to work with the spectrogram representation of audio and do not directly interact with the raw audio waveform. However, it's important to note that spectrogram-like features are designed based on perceptual evidence and may not capture all the variational factors present in audio data. This reliance on spectrograms could potentially limit the full potential of GGAN and GAAE. To address this limitation, I established objective 3, aligning with the secondary aim of this research.

To fulfill objective 3, I introduced two models: CosCovNN and VQCCM. These models can classify audio directly from raw waveform data and have the potential to replace the foundational CNN architecture of GGAN and GAAE. This would enable these models to operate directly on raw audio data. I leave this possibility open for future researchers to explore and extend this research.

I introduced CosCovNN, which incorporates learnable Cosine filters, resulting in a remarkable 77% reduction in parameters. Through extensive comparisons on Speech and Nonspeech datasets, I demonstrated CosCovNN's effectiveness in directly classifying audio from raw waveforms. Furthermore, I provided clear guidelines

for integrating cosine filters into existing CNN models, facilitating enhanced performance and reduced complexity. These insights open up promising avenues for future research, allowing for the exploration of diverse CNN architectures tailored for audio modelling.

Additionally, I proposed VQCCM, an advanced iteration of CosCovNN, featuring Vector Quantisation (VQ) and Memory layers. In evaluations across a range of datasets, including Speech Command, Speech Emotion, Acoustic Scenes, Musical Instrument, and Speaker ID, VQCCM consistently achieved state-of-the-art results and outperformed benchmarks in specific instances. The integration of VQ and memory layers substantially enhanced CosCovNN's performance. This innovation offers potential for broader application and exploration within various CNN architectures found in the existing literature. Future research endeavours can delve into these promising outcomes further.

In summary, this research has advanced the field of unsupervised disentangled representation learning and audio modelling. It has introduced innovative models, shed light on new avenues for research, and contributed to the broader landscape of machine learning. The potential and versatility of these models extend beyond the scope of this study, offering exciting opportunities for future investigations and applications.

6.4. Synthesis of Findings Across Models

This thesis introduced three primary models: Guided Generative Adversarial Neural Network (GGAN), Guided Adversarial Autoencoder (GAAE), and Cosine Convolutional Neural Network (CosCovNN)/Vector Quantised Cosine Convolutional Neural Network with Memory (VQCCM), each offering unique strengths and addressing specific aspects of representation learning and audio generation.

- **Comparison on Similar Datasets:** The performance of both GGAN and GAAE was thoroughly evaluated on speech and non-speech datasets, illustrating their capabilities in extracting disentangled representations from unlabelled audio data. GGAN excels in scenarios requiring task-specific representations, showing high accuracy even with minimal labelled data. Its focus on guided learning enables the partitioning of the latent space in a way that enhances model performance for specific tasks. However, GAAE

expands on GGAN's capabilities by offering broader applicability through its dual representation approach, which allows it to learn both task-specific and general representations. This dual functionality makes GAAE versatile, providing utility across a wider range of tasks, not limited to the conditions of the supervised signal.

- **Strengths and Limitations:** GGAN's primary strength lies in its efficient use of limited labelled data to guide the learning process, making it particularly useful in cases where acquiring labelled data is challenging. However, its reliance on guided learning within a specific latent space restricts its ability to generalise to unrelated tasks. GAAE addresses this limitation by learning generalised representations alongside task-specific ones, but this comes at the cost of increased computational demands due to the complexity of managing dual latent spaces. On the other hand, CosCovNN and VQCCM are tailored for direct audio classification from raw waveforms, bypassing the need for manually crafted features like spectrograms. This direct approach not only reduces dependency on feature engineering but also significantly enhances parameter efficiency. VQCCM, in particular, demonstrates superior performance across diverse audio datasets, setting new benchmarks in audio classification accuracy and robustness.

- **Insights:** Collectively, these findings highlight the complementary strengths of the proposed models. GGAN and GAAE excel in disentangled representation learning, making them ideal for tasks that benefit from understanding and utilising latent structure in the data. Meanwhile, the introduction of CosCovNN and VQCCM provides a critical advancement by directly processing raw waveforms, thus reducing the dependency on spectrograms and other hand-crafted features. This integration of direct waveform processing into the broader framework of representation learning offers a more streamlined and potentially more powerful approach to handling audio data. By bridging the gap between task-specific and general-purpose representations, these models contribute to a more holistic understanding and application of unsupervised learning techniques in audio processing.

6.5. Limitations of the Proposed Methods

Despite their strengths, the models presented in this thesis have limitations that must be acknowledged:

- **GGAN:** While GGAN is effective in learning task-specific representations from unlabelled data with minimal labelled supervision, its ability to generalise across unrelated tasks is constrained due to its focused partitioning of the latent space based on the specific guidance provided. This limitation restricts its applicability when the goal is to develop models that can handle a broader range of tasks without retraining.

- **GAAE:** GAAE addresses the generalisation challenge by introducing a dual representation space for both guided and general features. However, this dual approach necessitates more complex training procedures and significantly increases computational requirements. The need for higher computational resources could pose challenges for deploying GAAE in large-scale or real-time applications, where efficiency is a critical factor.

- **CosCovNN and VQCCM:** These models, designed for direct waveform processing, are efficient in terms of parameter usage and demonstrate strong performance in classification tasks. However, as relatively new approaches, they still require further validation across a broader spectrum of audio types and conditions, such as those involving environmental noise, diverse audio sources, or varying recording qualities. Establishing their robustness and versatility across different audio scenarios remains an essential step in their development.

6.6. Future Research Directions

To further advance the field of disentangled representation learning and audio processing, future research could explore:

- **Hybrid Approaches:** One promising avenue involves integrating GGAN and GAAE with more advanced generative models like diffusion models or leveraging transformer-based architectures. These hybrids could enhance the models' capacity to learn high-quality representations with reduced reliance on labelled data, potentially improving both the quality and flexibility of the representations learned.

- **Scalability Improvements:** Optimising the architecture of GAAE to reduce its computational footprint could make it more accessible for real-world applications. This might involve exploring more efficient training algorithms, incorporating techniques such as model pruning or quantisation, or even developing lightweight variants that retain the core strengths of the model.
- **Application to Broader Audio Types:** Expanding the evaluation of CosCovNN and VQCCM to include a wider variety of audio types, including challenging conditions like environmental noise or overlapping sound sources, could help validate their robustness. Understanding how these models perform under less controlled conditions would provide valuable insights into their potential real-world applications and guide further refinements.
- **Cross-Modal Representation Learning:** Extending these models to cross-modal tasks, such as linking audio to text or image data, could open new interdisciplinary research opportunities. This direction would involve adapting the models to handle and integrate multi-modal data, potentially leading to new applications in areas like multimedia analysis, content creation, and human-computer interaction.

REFERENCES

1. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* nature, 2015. **521**(7553): p. 436-444.
2. Das, K. and R.N. Behera, *A survey on machine learning: concept, algorithms and applications*. Vol. 5. 2017: International Journal of Innovative Research in Computer and Communication Engineering.
3. Zhong, G., L.-N. Wang, and X.L.J. Dong, *An overview on data representation learning: From traditional feature learning to recent deep learning*. Vol. 2. 2016: The Journal of Finance and Data Science.
4. Chen, Y., et al., *Minimalistic unsupervised representation learning with the sparse manifold transform*. 2022: The Eleventh International Conference on Learning Representations.
5. Radford, A., L. Metz, and S. Chintala, *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2015: International Conference on Learning Representations
6. Lee, H., et al., *Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations*. 2009: 26th annual international conference on machine learning, ACM.
7. Goodfellow, I., Y. Bengio, and A. Courville, *Deep Learning*. 2016: MIT Press.
8. Bengio, Y., A. Courville, and P. Vincent, *Representation learning: A review and new perspectives*. Vol. 35. 2013: IEEE transactions on pattern analysis and machine intelligence.
9. Locatello, F., et al., *Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations*. 2019: International Conference on Machine Learning.
10. Spurr, A., E. Aksan, and O. Hilliges, *Guiding infogan with semi-supervision*. 119--134: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer.
11. Springenberg and J. Tobias, *Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks*, in *4th International Conference on Learning Representations*. 2016.
12. Lucic, M., et al., *High-fidelity image generation with fewer labels*. 2019, arXiv preprint arXiv:1903.02271.
13. Goodfellow, I., et al., *Generative adversarial nets.* Advances in neural information processing systems, 2014: p. 2672--2680.
14. Karras, T., et al., *Progressive Growing of GANs for Improved Quality, Stability, and Variation.* International Conference on Learning Representations (ICLR), 2018.
15. Karras, T., S. Laine, and T. Aila, *A style-based generator architecture for generative adversarial networks*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019. p. 4401--4410.

16. Brock, A., J. Donahue, and K. Simonyan, *Large Scale GAN Training for High Fidelity Natural Image Synthesis.* International Conference on Learning Representations (ICLR), 2019.
17. Donahue, J. and K. Simonyan, *Large Scale Adversarial Representation Learning.* Advances in Neural Information Processing Systems}, 2019: p. 10541--10551.
18. Marafioti, A., et al., *Adversarial generation of time-frequency features with application in audio synthesis.* International Conference on Machine Learning, 2019: p. 4352--4362.
19. Engel, J., et al., *GANSynth: Adversarial Neural Audio Synthesis.* International Conference on Learning Representations (ICLR), 2019.
20. Pruvsa, Z., et al., *The Large Time-Frequency Analysis Toolbox 2.0*, in *Sound, Music, and Motion*. 2014, Springer International Publishing. p. 419--442.
21. Bengio, Y., et al., *Generalized Denoising Auto-encoders As Generative Models.* 2013.
22. Ravanelli, M. and Y. Bengio, *Speaker recognition from raw waveform with sincnet.* IEEE Spoken Language Technology Workshop (SLT), 2018: p. 1021--1028.
23. Jung, J.-w., et al., *Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification.* Interspeech, 2019: p. 1268--1272.
24. Carrio, A., et al., *A review of deep learning methods and applications for unmanned aerial vehicles.* Journal of Sensors, 2017. **2017**.
25. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *Imagenet classification with deep convolutional neural networks.* Advances in neural information processing systems, 2012. **25**.
26. Barrault, L., et al., *SeamlessM4T-Massively Multilingual & Multimodal Machine Translation.* arXiv preprint arXiv:2308.11596, 2023.
27. Singh, S.P., et al. *Machine translation using deep learning: An overview*. in *2017 international conference on computer, communications and electronics (comptelix)*. 2017. IEEE.
28. Yang, S., Y. Wang, and X. Chu, *A survey of deep learning techniques for neural machine translation.* arXiv preprint arXiv:2002.07526, 2020.
29. Malik, M., et al., *Automatic speech recognition: a survey.* Multimedia Tools and Applications, 2021. **80**: p. 9411-9457.
30. Nassif, A.B., et al., *Speech recognition using deep neural networks: A systematic review.* IEEE access, 2019. **7**: p. 19143-19165.
31. Noda, K., et al., *Audio-visual speech recognition using deep learning.* Applied intelligence, 2015. **42**: p. 722-737.
32. Yenduri, G., et al., *Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions.* arXiv preprint arXiv:2305.10435, 2023.
33. Devlin, J., et al., *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv preprint arXiv:1810.04805, 2018.

34. Ghandi, T., H. Pourreza, and H. Mahyar, *Deep learning approaches on image captioning: A review.* ACM Computing Surveys, 2023. **56**(3): p. 1-39.
35. Hossain, M.Z., et al., *A comprehensive survey of deep learning for image captioning.* ACM Computing Surveys (CsUR), 2019. **51**(6): p. 1-36.
36. Sharma, H., et al. *Image captioning: a comprehensive survey*. in *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*. 2020. IEEE.
37. Amaresh, M. and S. Chitrakala. *Video captioning using deep learning: an overview of methods, datasets and metrics*. in *2019 International Conference on Communication and Signal Processing (ICCSP)*. 2019. IEEE.
38. Xu, J., et al. *Learning multimodal attention LSTM networks for video captioning*. in *Proceedings of the 25th ACM international conference on Multimedia*. 2017.
39. Zheng, C., et al., *Deep learning-based human pose estimation: A survey.* ACM Computing Surveys, 2023. **56**(1): p. 1-37.
40. Liu, Z., et al. *Swin transformer: Hierarchical vision transformer using shifted windows*. in *Proceedings of the IEEE/CVF international conference on computer vision*. 2021.
41. Alexey, D., *An image is worth 16x16 words: Transformers for image recognition at scale.* arXiv preprint arXiv: 2010.11929, 2020.
42. Yao, T., et al., *Dual vision transformer.* IEEE transactions on pattern analysis and machine intelligence, 2023. **45**(9): p. 10870-10882.
43. Song, B., Y. Wu, and Y. Xu. *ViTCN: Vision Transformer Contrastive Network For Reasoning*. in *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*. 2024. IEEE.
44. Esser, P., R. Rombach, and B. Ommer. *Taming transformers for high-resolution image synthesis*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.
45. Ho, J., et al., *Imagen video: High definition video generation with diffusion models.* arXiv preprint arXiv:2210.02303, 2022.
46. Ho, J., et al., *Cascaded diffusion models for high fidelity image generation.* The Journal of Machine Learning Research, 2022. **23**(1): p. 2249-2281.
47. Chan, H.-P., et al., *Deep learning in medical image analysis.* Deep Learning in Medical Image Analysis: Challenges and Applications, 2020: p. 3-21.
48. Chen, X., et al., *Recent advances and clinical applications of deep learning in medical image analysis.* Medical Image Analysis, 2022. **79**: p. 102444.
49. Shen, D., G. Wu, and H.-I. Suk, *Deep learning in medical image analysis.* Annual review of biomedical engineering, 2017. **19**: p. 221-248.

50. Santos, I., et al., *Artificial neural networks and deep learning in the visual arts: A review.* Neural Computing and Applications, 2021. **33**: p. 121-157.
51. Agostinelli, A., et al., *Musiclm: Generating music from text.* arXiv preprint arXiv:2301.11325, 2023.
52. Copet, J., et al., *Simple and Controllable Music Generation.* arXiv preprint arXiv:2306.05284, 2023.
53. Kreuk, F., et al., *Audiogen: Textually guided audio generation.* arXiv preprint arXiv:2209.15352, 2022.
54. Nijkamp, E., et al., *Codegen: An open large language model for code with multi-turn program synthesis.* arXiv preprint arXiv:2203.13474, 2022.
55. Roziere, B., et al., *Code llama: Open foundation models for code.* arXiv preprint arXiv:2308.12950, 2023.
56. Oquab, M., et al. *Learning and transferring mid-level image representations using convolutional neural networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
57. Russakovsky, O., et al., *Imagenet large scale visual recognition challenge.* International journal of computer vision, 2015. **115**: p. 211-252.
58. Zhou, B., et al., *Learning deep features for scene recognition using places database.* Advances in neural information processing systems, 2014. **27**.
59. Long, J., E. Shelhamer, and T. Darrell. *Fully convolutional networks for semantic segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
60. Karpathy, A. and L. Fei-Fei. *Deep visual-semantic alignments for generating image descriptions*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
61. Diment, A. and T. Virtanen. *Transfer learning of weakly labelled audio*. in *2017 ieee workshop on applications of signal processing to audio and acoustics (waspaa)*. 2017. IEEE.
62. Triantafyllopoulos, A. and B.W. Schuller. *The role of task and acoustic similarity in audio transfer learning: Insights from the speech emotion recognition case*. in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021. IEEE.
63. Van Den Oord, A., S. Dieleman, and B. Schrauwen. *Transfer learning by supervised pre-training for audio-based music classification*. in *Conference of the International Society for Music Information Retrieval (ISMIR 2014)*. 2014.
64. Yosinski, J., et al., *How transferable are features in deep neural networks?* Advances in neural information processing systems, 2014. **27**.

65. Klementiev, A., I. Titov, and B. Bhattarai. *Inducing crosslingual distributed representations of words*. in *Proceedings of COLING 2012*. 2012.
66. Gouws, S., Y. Bengio, and G. Corrado. *Bilbowa: Fast bilingual distributed representations without word alignments*. in *International Conference on Machine Learning*. 2015. PMLR.
67. Larochelle, H., D. Erhan, and Y. Bengio. *Zero-data learning of new tasks*. in *AAAI*. 2008.
68. Palatucci, M., et al., *Zero-shot learning with semantic output codes.* Advances in neural information processing systems, 2009. **22**.
69. He, K., et al. *Masked autoencoders are scalable vision learners*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.
70. Radford, A., et al. *Learning transferable visual models from natural language supervision*. in *International conference on machine learning*. 2021. PMLR.
71. Zhang, B., et al., *Long-clip: Unlocking the long-text capability of clip.* arXiv preprint arXiv:2403.15378, 2024.
72. Fan, L., et al., *Improving clip training with language rewrites.* Advances in Neural Information Processing Systems, 2024. **36**.
73. McCarthy, J. and P.J. Hayes, *Some philosophical problems from the standpoint of artificial intelligence*, in *Readings in artificial intelligence*. 1981, Elsevier. p. 431-450.
74. Botteghi, N., M. Poel, and C. Brune, *Unsupervised representation learning in deep reinforcement learning: A review.* arXiv preprint arXiv:2208.14226, 2022.
75. Abukmeil, M., et al., *A survey of unsupervised generative models for exploratory data analysis and representation learning.* Acm computing surveys (csur), 2021. **54**(5): p. 1-40.
76. Floridi, L. and M. Chiriatti, *GPT-3: Its nature, scope, limits, and consequences.* Minds and Machines, 2020. **30**: p. 681-694.
77. Touvron, H., et al., *Llama 2: Open foundation and fine-tuned chat models.* arXiv preprint arXiv:2307.09288, 2023.
78. Grill, J.-B., et al., *Bootstrap your own latent-a new approach to self-supervised learning.* Advances in neural information processing systems, 2020. **33**: p. 21271-21284.
79. Baevski, A., et al. *Data2vec: A general framework for self-supervised learning in speech, vision and language*. in *International Conference on Machine Learning*. 2022. PMLR.
80. Liu, C., et al. *Bootstrapping Large Language Models for Radiology Report Generation*. in *Proceedings of the AAAI Conference on Artificial Intelligence*. 2024.
81. Gao, Y., et al., *Latent representation discretization for unsupervised text style generation.* Information Processing & Management, 2024. **61**(3): p. 103643.
82. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning Internal Representations by Error Propagation, Parallel Distributed*

Processing, Explorations in the Microstructure of Cognition, ed. DE Rumelhart and J. McClelland. Vol. 1. 1986. Biometrika, 1986. **71**: p. 599-607.
83. Hinton, G.E., S. Osindero, and Y.-W. Teh, *A fast learning algorithm for deep belief nets.* Neural computation, 2006. **18**(7): p. 1527-1554.
84. Hinton, G.E. and R.R. Salakhutdinov, *Reducing the dimensionality of data with neural networks.* science, 2006. **313**(5786): p. 504-507.
85. Erhan, D., et al. *The difficulty of training deep architectures and the effect of unsupervised pre-training*. in *Artificial intelligence and statistics*. 2009. PMLR.
86. Erhan, D., et al. *Why does unsupervised pre-training help deep learning?* in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 2010. JMLR Workshop and Conference Proceedings.
87. Salakhutdinov, R. and G. Hinton. *Deep boltzmann machines*. in *Artificial intelligence and statistics*. 2009. PMLR.
88. Pearson, K., *LIII. On lines and planes of closest fit to systems of points in space.* The London, Edinburgh, and Dublin philosophical magazine and journal of science, 1901. **2**(11): p. 559-572.
89. Sirovich, L. and M. Kirby, *Low-dimensional procedure for the characterization of human faces.* Josa a, 1987. **4**(3): p. 519-524.
90. Fisher, R.A., *The use of multiple measurements in taxonomic problems.* Annals of eugenics, 1936. **7**(2): p. 179-188.
91. Schölkopf, B., A. Smola, and K.-R. Müller, *Nonlinear component analysis as a kernel eigenvalue problem.* Neural computation, 1998. **10**(5): p. 1299-1319.
92. Baudat, G. and F. Anouar, *Generalized discriminant analysis using a kernel approach.* Neural computation, 2000. **12**(10): p. 2385-2404.
93. Yan, S., et al., *Graph embedding and extensions: A general framework for dimensionality reduction.* IEEE transactions on pattern analysis and machine intelligence, 2006. **29**(1): p. 40-51.
94. Zhong, G., Y. Chherawala, and M. Cheriet. *An empirical evaluation of supervised dimensionality reduction for recognition*. in *2013 12th International Conference on Document Analysis and Recognition*. 2013. IEEE.
95. Coates, A., A. Ng, and H. Lee. *An analysis of single-layer networks in unsupervised feature learning*. in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 2011. JMLR Workshop and Conference Proceedings.
96. Coates, A. and A.Y. Ng, *Learning feature representations with k-means*, in *Neural Networks: Tricks of the Trade: Second Edition*. 2012, Springer. p. 561-580.
97. Zhong, G., et al., *An overview on data representation learning: From traditional feature learning to recent deep learning.* The Journal of Finance and Data Science, 2016. **2**(4): p. 265-278.

98. Alain, G. and Y. Bengio, *What regularized auto-encoders learn from the data-generating distribution.* The Journal of Machine Learning Research, 2014. **15**(1): p. 3563-3593.
99. Bengio, Y., et al., *Generalized denoising auto-encoders as generative models.* Advances in neural information processing systems, 2013. **26**.
100. Glorot, X., A. Bordes, and Y. Bengio. *Domain adaptation for large-scale sentiment classification: A deep learning approach*. in *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011.
101. Vincent, P., et al., *Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.* Journal of machine learning research, 2010. **11**(12).
102. Zhao, J., et al., *Stacked what-where auto-encoders.* arXiv preprint arXiv:1506.02351, 2015.
103. Rifai, S., et al. *Higher order contractive auto-encoder*. in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II 22*. 2011. Springer.
104. Zhang, R., P. Isola, and A.A. Efros. *Split-brain autoencoders: Unsupervised learning by cross-channel prediction*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
105. Kingma, D.P. and M. Welling, *Auto-encoding variational bayes.* arXiv preprint arXiv:1312.6114, 2013.
106. Kingma, D.P., et al., *Improved variational inference with inverse autoregressive flow.* Advances in neural information processing systems, 2016. **29**.
107. Larochelle, H. and I. Murray. *The neural autoregressive distribution estimator*. in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 2011. JMLR Workshop and Conference Proceedings.
108. Makhzani, A., *Unsupervised representation learning with autoencoders*. 2018: University of Toronto (Canada).
109. Makhzani, A., et al., *Adversarial autoencoders.* arXiv preprint arXiv:1511.05644, 2015.
110. Makhzani, A. and B.J. Frey, *Pixelgan autoencoders.* Advances in Neural Information Processing Systems, 2017. **30**.
111. Chen, X., et al., *Infogan: Interpretable representation learning by information maximizing generative adversarial nets.* Advances in neural information processing systems, 2016. **29**.
112. Donahue, J., P. Krähenbühl, and T. Darrell, *Adversarial feature learning.* arXiv preprint arXiv:1605.09782, 2016.
113. Arik, S., et al., *Neural voice cloning with a few samples.* Advances in neural information processing systems, 2018. **31**.

114. Donahue, C., J. McAuley, and M. Puckette, *Synthesizing audio with generative adversarial networks.* arXiv preprint arXiv:1802.04208, 2018. **1**.
115. Wan, C.-H., S.-P. Chuang, and H.-Y. Lee. *Towards audio to scene image synthesis using generative adversarial network*. in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019. IEEE.
116. Kong, Z., et al., *Diffwave: A versatile diffusion model for audio synthesis.* arXiv preprint arXiv:2009.09761, 2020.
117. Grassucci, E., et al. *Diffusion models for audio semantic communication*. in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2024. IEEE.
118. Chen, N., et al., *Wavegrad 2: Iterative refinement for text-to-speech synthesis.* arXiv preprint arXiv:2106.09660, 2021.
119. Oord, A.v.d., Y. Li, and O. Vinyals, *Representation learning with contrastive predictive coding.* arXiv preprint arXiv:1807.03748, 2018.