VICTORIA UNIVERSITY OF WELLINGTON
*Te Whare Wānanga o te Ūpoko o te Ika a Māui*


School of Engineering and Computer Science
*Te Kura Mātai Pūkaha, Pūrorohiko*

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

# Hierarchical Audio-Conditional Image Generation with AudioCLIP Latents

Richard Kyle

Supervisors: Andrew Lensen, Stephen Marsland

Submitted in partial fulfilment of the requirements for
Master of Artificial Intelligence.

**Abstract**

This research project explores the machine learning domain of Audio-to-Image generation, a field that employs audio signals to condition the image generation process. Existing approaches to this problem are limited by low resolution outputs and highly task specific applications. However, breakthroughs in the analogous field of Text-to-Image generation offer promising avenues to address these challenges. We present a proof-of-concept study demonstrating how integrating these advancements into an Audio-to-Image generation framework can lead to substantial performance improvements. Specifically, we detail the first implementation of both a diffusion image generator and the advanced embedding network, AudioCLIP, within the Audio-to-Image generation domain. Results show a significant 70% improvement in FID scores in comparison to current methodologies of equivalent image resolutions. We accompany this with a series of experimental analyses highlighting how effective training strategies, hyper-parameter configurations, and network architectures can be used to optimise this framework. By doing so, we pave the way for future research directions that could further advance Audio-to-Image generation models.

# Contents

# Figures

# Chapter 1

# Introduction

The field of artificial intelligence has recently experienced a period of rapid advancements. Much of this progress has been driven by the core sub-field of machine learning. Machine learning involves the development of algorithms that can learn from data [8]. These learnings can then be applied to new scenarios, helping solve a diverse array of complex problems [9].

Generative modelling is a uniquely interesting aspect of machine learning. It focuses on creating new data that could convincingly belong to a specific, real data distribution [10]. For instance, a generative model could be trained on a large dataset of modern artworks, after which, it could be used to create new works in the style of Picasso, Dali or Monet.

This contrasts with discriminative modeling, which separates existing data into categories to solve problems [10]. While discriminative models learn boundaries between groups of data, generative models learn the distribution from which data arises [10]. Once an appropriate distribution has been learnt, it can be sampled from, facilitating the creation of new instances.

## 1.1 Problem Statement

Audio-to-Image generation is a fascinating area of generative modeling. Work in this domain employs audio signals to condition the image generation process [2]. This can be understood as a multi-modal problem, where information from the audio modality is translated into the image modality. Previous research has effectively demonstrated the ability to use bird calls to generate corresponding images of birds [11], and sounds from musical performances to generate imagery of source instruments [2].

This is achieved by employing a two sub-network model architecture. An audio encoding network is first used to convert an audio sample into an embedding. In this context, embeddings can be understood as learned numerical representations which describe the content of an audio sample [2]. The embedding is then passed to an image generation network where it is used to guide the generative process. Done effectively, the generated output image will visually align with the audio signal.

## 1.2 Motivation

From a theoretical perspective, Audio-to-Image generation possesses an important quality. Namely, it is the first formulation of an inter-sensory multi-modal image generation problem [2]. This is particularly interesting for two reasons. Firstly, combining inter-sensory information is a fundamentally how we interact with the world. As humans, we frequently

combine information from multiple sensory channels to enrich our understanding of the scenarios we find ourselves in. As such, it is reasonable to believe that effectively integrating this capacity into machine learning models could prove highly beneficial.

This leads into the second desirable quality. To effectively integrate multi-sensory information into machine learning models, it is beneficial to understand how translation between the modalities occurs. Audio-to-Image generation affords us with this capacity. By providing a visual output, we gain an examinable means of understanding the inter-modal relationships between audio and images. This has the potential to provide effective feedback to researchers, that could be used to help improve future multi-sensory networks.

To realize these benefits, it is important to develop high performing Audio-to-Image generation models. By doing so, we gain a solid foundation from which to conduct further research. This said, Audio-to-Image generation is a relatively underexplored domain, with state-of-the-art models limited to low resolution outputs and highly task-specific applications [2][11].

Interestingly, the related domain of Text-to-Image generation is substantially more advanced. Within the past 12 months, models such as Stable Diffusion [12], Imagen [13], and DALL-E 2 [14] have demonstrated an impressive ability to generate high-quality, semantically aligned images, in a broad array of contexts.

These developments are the result of a variety of factors. A number of which are beyond the scope of this research project to consider incorporating. Specifically, the compilation of large-scale datasets and application of extensive computational resources [14][13][12][15]. That said, there are two architectural improvements we have the capability to investigate. Those being diffusion networks [16] and highly visually expressive contrastively trained embedding networks [17][14]. The former provides a tractable means of generating images that can be effectively applied to large scale training configurations [16]. The latter affords us with a practical method of encoding visually meaningful information into embeddings [17], a technique associated with the improvements in the visual quality of images [14].

## 1.3 Goals

In this research project, we focus our efforts upon incorporating developments from the Text-to-Image generation domain into the Audio-to-Image domain. We specifically do the following:

- Conduct a comprehensive literature review, building off the author's AIML 501 project. Here we provide information on foundational concepts, detail existing literature in the Audio-to-Image generation domain, and present relevant details from the Text-to-Image generation domain.

- Propose the first application of a diffusion model in the Audio-to-Image generation domain.

- Integrate a highly visually expressive embedding network into the Audio-to-Image generation domain.

- Provide experimental analyses centered upon improving the performance of both networks.

- Evaluate the performance of both networks in comparison to existing literature.

## 1.4 Organization

We arrange this research project into six distinct chapters. A literature review is provided in Chapter 2, detailing important foundational concepts and highlighting pertinent developments in Audio-to-Image generation and related domains. We note that some of this literature review is taken from the author's AIML 501 project completed last semester. A table is presented at the beginning of Chapter 2 identifying which aspects of the review come from our previous work. In Chapter 3, we detail the proposed model architecture, describe its implementation and provide justification for our decisions. Chapter 4, Experimental Design, provides detailed descriptions of the experimental analyses conducted. The results of these experiments are presented in Chapter 5, alongside a discussion of their relevance. Finally, we conclude with Chapter 6, summarizing key findings, limitations and suggesting directions for future work.

# Chapter 2

# Literature Review

## 2.1 Chapter Overview

In this chapter we provide a representative literature review to support this research project. Here, we detail the foundational machine learning concepts necessary to engage with the topic, alongside relevant works within the Audio-to-Image and Text-to-Image generation domains.

It is important to mention that certain aspects of this literature review have been adapted from the author's AIML 501 project completed last semester. We delineate these sections in Table 2.1. Those parts taken from AIML 501 are given with a level of adaptation. This captures the effort required to translate each section into an appropriate format for the current report.

Table 2.1: Table of work origin. Adaptation defines how much work went into translating information from AIML 501 to 589.

| Section | Source | Adaptation Effort |
|---|---|---|
| Deep Neural Networks | 501 | Low |
| Convolutional Neural Networks | 589 | N/A |
| Attention | 589 | N/A |
| Generative Adversarial Networks | 501 | Low |
| Conditional Generative Adversarial Networks | 501 | Low |
| Denoising Diffusion Models | 501 | Moderate |
| Audio Encoders | 589 | N/A |
| Inception Score (IS) | 501 | Low |
| Frechet Inception Distance (FID) | 501 | Low |
| Image Classifier Networks | 589 | N/A |
| Deep Cross Modal Audio Visual Generation | 589 | N/A |
| CMCGAN | 589 | N/A |
| Cross Modal Contrastive Representation Learning | 589 | N/A |
| AudioCLIP | 589 | N/A |
| Audio-to-Image Summary | 589 | N/A |
| GLIDE | 501 | Moderate |
| DALL-E 2 | 501 | Moderate |
| Latent Diffusion Models | 501 | Moderate |
| Summary | 589 | N/A |

This literature review is organized into five sections. Section one details the foundational concepts our research project is predicated upon. This includes an overview of deep neural networks, convolutional networks, and attention mechanisms. With these understood, we transition into a discussion of relevant model architectures. Specifically, describing the functioning of Generative Adversarial Networks (GANs), diffusion models, and audio encoders. GANs are commonly used in current Audio-to-Image generation research, while diffusion models are the generative backbone of our proposed advancement. Additionally, audio encoders serve a fundamental role, creating the conditioning information used to guide the generative process. Consequently, it is necessary for readers to gain an understanding for how each of these networks function.

We then provide a brief discussion of relevant performance metrics, followed by an overview of pertinent literature in the Audio-to-Image generation domain. It is beyond the scope of this research project to provide a comprehensive discussion of Audio-to-Image generation literature, thus we only highlight those works that are most relevant to the current project. Finally, we end with a review of contributions within the Text-to-Image generation domain. By doing so, we aim to provide readers with an understanding of where our contributions were derived from.

## 2.2 Foundational Concepts

### 2.2.1 Deep Neural Networks

At their core, DNNs stack layers of computations sequentially, learning a mapping function $f(\cdot)$ that converts a set of inputs $\mathbf{x}$ into desired outputs $\mathbf{y}$ [9]. This computation is defined by the networks parameters $\theta$. Parameters are learnt during training via the optimization of a loss function. Loss functions measure how well a network maps $\mathbf{x}$ to $\mathbf{y}$. By taking the gradient of a loss function, with respect to $\theta$, parameters can be adjusted to improve the mapping function. Repeating this process iteratively allows DNNs to learn effective mappings, facilitating the solving of complex problems [9].

Deep generative networks (DGNs) are a specific type of DNN that can *create* new data [10]. These models learn a mapping function to convert input data to a specific, sampleable distribution [9]. For instance, a DGN may learn to map random noise to a distribution of modern artworks $p(\mathbf{x})$. Once this has been learnt, the model can be given samples of noise and convert them into new instances that mimic modern art.

### 2.2.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) are specifically designed networks that specialize in processing data structured in an inherently meaningful grid, most commonly images [9]. To achieve this, CNNs detect spatial patterns by employing three types of computational layers.

#### Convolutional Layers

Convolutional layers scan filters sequentially across an input [9]. Filters detect patterns within the input, for instance, the presence of edges, corners and colours etc. Each filter is comprised of multiple kernels that act on individual channels of the input [9]. Each kernel tends to be a few pixels wide in both the horizontal and vertical direction, and is parameterized by a set of learnable weights. These weights determine the patterns each kernel learns to detect, and are optimized during training in the same fashion described in Section 2.2.1.

The output of a convolution layer can be understood as a representation of the presence of those patterns within the input.

#### Pooling Layers

Pooling layers are used to reduce the dimensionality of the input. Doing so lessens the computational effort required to train the model, and prevents overfitting [9]. They achieve this by synthesizing information within a defined range of the input. For instance they may scan a $2 \times 2$ pixel square sequentially over the output of a convolutional layer, at each point taking only the maximum value within this range as the output. This approach is known as max pooling and is commonly applied [9]. However, other approaches (e.g. average pooling) can also be implemented [9].

#### Fully Connected Layers

Finally, CNNs employ fully connected layers to convert feature representations into outputs [9]. These are used to synthesize the features produced via convolutional and pooling layers into a predefined form. For instance, a fully connected layer may be employed to convert convolutional outputs into 2 values. These could represent the models confidence the original input to the network was an image of either a cat or a dog. By performing this function, the fully connected layer allows the CNN to apply its learnt features to an arbitrary array of classification problems.

### 2.2.3 Attention

Taking inspiration from human cognition, attention mechanisms allow neural networks to focus on different portions of the input, proportional to their relevance [18]. This feature has proven highly effective in machine translation [19] and computer vision tasks [20], enabling the more efficient and effective processing of complex data.

We can separate attention operations into two categories, those being self-and-cross-attention. The former calculation works by taking an input $X$ and linearly projecting it into query, key and value vectors $Q, K$ and $V$. Each vector serves a different role in the attention calculation, thus we transform the input differently for each. Transformations are achieved linearly by multiplying $X$ by three *different* learnable weight matrices $W_q, W_k$ and $W_v$ such that $Q = X \cdot W_q$, $K = X \cdot W_k$ and $V = X \cdot W_v$ [18].

Once complete, we use query and key vectors to calculate attention scores. We can understand the query vector as representing the original input, and the key vector as representing the relevance of each aspect of the input. Attention scores are derived by taking the dot product between $Q$ and $K$ denoted by $QK^T$[18]. Performing this operation allows us to calculate the similarity between elements of each vector, with higher values indicating a greater degree of similarity [18].

We then normalize attention scores by dividing them by the square-root of the dimensionality of $K$. This serves to prevent exploding/vanishing gradients during the subsequent softmax operation by avoiding overly large/small attention scores [18]. Finally, we multiply $V$ by the softmaxed attention scores, producing a transformed version of the input, whereby values of its most relevant components are increased. The formula for the attention calculation can be seen below in Equation 2.1.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

Cross-attention operates similarly to self-attention; however, $W_k$ and $W_v$ are utilized to project another vector $Y$ into $K$ and $V$ [12]. In this configuration, attention scores represent the relevance between each element of $X$ with respect to a secondary vector $Y$. This is especially helpful when the goal is to modify $X$ to emphasize values related to $Y$. For example, imagine $X$ is a part of an image, and $Y$ is a textual condition. Implementing cross-attention between $X$ and $Y$, would then provide a mechanism to adjust the image to better align with conditioning information [12].

## 2.3 Model Architecture

### 2.3.1 Generative Adversarial Networks

Although Generative Adversarial Networks (GANs) are not an aspect of the contributions provided by this research project, they do play a significant role within the existing Audio-to-Image generation literature. Consequently, a succinct overview is provided below.

Introduced by Goodfellow et al. [21] in 2014, GANs are composed of two opposing networks trained simultaneously. The basic structure involves training a generator $G$ to convert noise $z$ sampled from a normal distribution $p_z$ into a target data distribution $P_{data}$. The generator's outputs are then fed to a discriminator $D$, which is trained to distinguish generated data from real data $x \sim P_{data}$. This training process can be understood as a competitive game between the two networks, with performance measured by the value function $V(G, D)$.

$$V(G, D) = \min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[1 - \log D(G(z))]. \tag{2.2}$$

Specifically, the discriminator $D$ is trained to maximize the log-likelihood of assigning the correct label to inputs ($\log D(x)$ for real inputs and $1 - \log D(G(z))$ for generated inputs). In contrast, the generator $G$ aims to minimize the log-likelihood of the discriminator detecting its output as false ($1 - \log D(G(z))$) [21]. It is worth mentioning that minimizing the objective for $G$ can lead to unstable gradients [22], so in practice, the generator's objective is often redefined to maximizing $\log D(G(z))$.

GANs are noteworthy for not possessing an explicit likelihood function over the data distribution [21]. Instead they optimize the value function. This can be understood as an implicit likelihood function, where the data distribution is learnt by proxy of learning to fool the discriminator. In other words, the generator never gets explicit access to the likelihood function of the data. This can make training GANs challenging, as there is nothing forcing the generator to learn the desired data distribution [23].

### 2.3.2 Conditional GANs

By incorporating a conditioning variable $y$ into the value function, GANs can be redefined as conditional generative models [24]. Under this implementation, both models $G$ and $D$ are provided with an additional input $y$. In the image generation case, $y$ could specify the content images ought to contain. For instance, $y$ could represent different types of animals, if $y = dog$, generated outputs would have to reflect this. This provides users with more control over the generative process. The value function for the conditional case is defined as:

$$V(G, D) = \min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x|y)] + \mathbb{E}_{z \sim p_z}[1 - \log D(G(z|y))]. \tag{2.3}$$

Audio-to-Image models condition the generative process on audio samples [2]. In this case $y$ can be understood as an audio embedding. We provide more detail on this specific interpretation in Section 2.3.4.

### 2.3.3 Denoising Diffusion Models

Diffusion models serve as the architectural foundation for many state-of-the-art Text-to-Image generation models [15][14][12][13], making them a central focus of this research project.

First introduced by Sohl-Dickstein et al. [16] in 2015, diffusion models offer a highly tractable generative alternative to GANs, and consist of two primary processes. The forward noising process involves iteratively adding noise to data, gradually eroding its structure [16]. By doing so, the output approaches the distribution from which noise was sampled [25]. The reverse process undoes this operation, iteratively removing noise and reinstating the structure of the original distribution [16]. If an effective reverse process is learnt, the model will be able to reproduce the original data from samples of noise [16].

Both the forwards and reverse processes are Markov chain operations, where the output at time $t$ relies only on the output at the preceding time-step $t-1$ [26]. Thus, the forward process is defined as $q(x_t|x_{t-1})$ and the reverse process as $p_\theta(x_{t-1}|x_t)$, where $x_t$ represents the original data after $t$ noising operations. Assuming noise follows an isotropic Gaussian distribution, as is most common, $q(x_t|x_{t-1})$ is defined in accordance with Equation 2.4.

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}). \tag{2.4}$$

Here, $\beta_t$ determines the variance and is scheduled to be relatively small when compared to the original data. This ensures the reverse process remains tractable [16]. The reverse process is defined as per Equation 2.5, where $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are neural networks that take noised data $x_t$ and time $t$ as inputs, and return parameter estimates for the mean and variance of the data distribution at time $t-1$.

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2.5}$$

While Nichol and Dhariwal [25] found it beneficial to use $\Sigma_\theta(x_t, t)$ to predict variance, many applications prefer to use a constant ($\beta\mathbf{I}$) for simplicity. This approach reduces computational requirements by eliminating the need for a secondary network. It is also worth noting that Ho et al. [26] found that estimating the noise added to the image at each time step was more effective than directly predicting mean values of previous image ($x_{t-1}$). Both approaches aim to produce $x_{t-1}$ as the desired output, although they employ different strategies to achieve this. As such, they can be viewed as distinct interpretations of the same problem [26].

U-Nets are widely used to learn the reverse process. They function by iteratively down-sampling an image $x_t$ into a bottleneck layer and employ up-sample blocks to generate the image from the preceding time-step, $x_{t-1}$ [26]. The standard architecture applies convolutions prior to up-and-down-sampling operations to extract feature representations from $x_t$. Additionally, layers of attention are incorporated between each up-and-down-sample block to facilitate an effective reverse process [26]. A visualization of the U-Net design can be seen in Figure 2.1.

When discussing diffusion models in the context of conditional image generation, it is essential to touch upon classifier and classifier-free guidance. These two techniques are frequently employed to promote improved alignment between generated images and conditioning information [27][28].

Classifier guidance utilizes a pre-trained classification network during image generation. At a given time-step, the diffusion model generates an image $x_t$, and the classifier is used

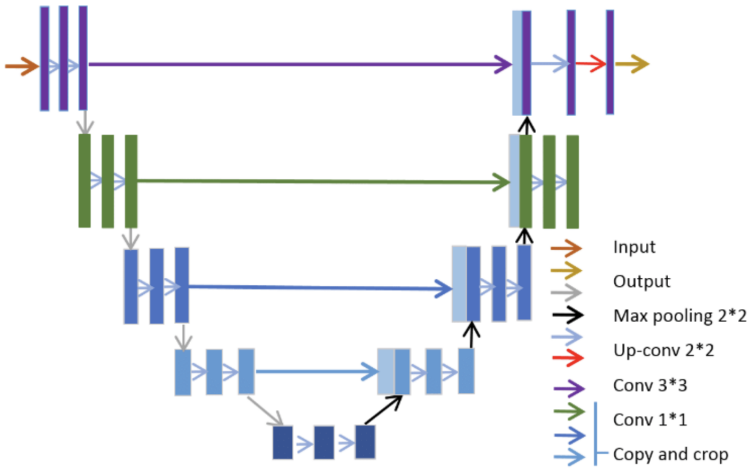



Figure 2.1: U-Net architecture. Image taken from [1].

to predict the class it believes $x_t$ belongs to. The gradient of the classifier's prediction with respect to the desired class provides information on how the image should be altered to better align with conditioning information [27]. The diffusion model then integrates this information to refine the image generation process [27].

Classifier-free guidance achieves the same goal without the need for an additional classification network [28]. During training conditional information is removed $p$ proportion of the time. By doing so, the generative network learns to produce both conditional and unconditional images. At sampling time, the model produces both a conditional and an unconditional image. By interpolating between these images, one can ascertain how the unconditional image ought to be altered to align it with conditioning information. By adding this difference to the unconditional image and scaling it by some value $s \geq 1$, the resulting image has been shown to better align with conditioning information [28].

### 2.3.4 Audio Encoders

Conditioning image generation on audio requires translating auditory information into a format machine learning models can understand. Audio encoding networks are used to facilitate this. These networks take audio samples as input, and convert them into $n$-dimensional numerical embedding vectors that represent their meaning.

It can sometimes be useful to understand embeddings vectors spatially. When $n = 2$ embeddings can be visualized on a scatterplot. Assuming the network has learnt effective representations, you could expect to see distinct clusters of audio signals located across the scatterplot. For example, the sound of car horns may be found in the upper-right section, while the quiet hum of a bee flying may be placed far down in the lower-left most corner. In practice, embeddings are frequently high dimensional objects that cannot be visualized, however, the principal of spatial separation remains the same.

Most audio encoding models operate on a time-frequency representation of the audio signal [6][2][3]. Doing so has proven to be an effective means of extracting useful information, and has been shown to produce better results than using the raw audio waveform [29]. A variety of time-frequency transformations have been applied within the literature, with the log-mel spectrogram transformation emerging as one of the most popular [2][3]. Log-mel spectrograms display audio signals as two-dimensional arrays where the $x$ axis represents time, and the $y$ frequency. The power within a short time window for each fre-

quency is then mapped to a colour scale, producing a visually interpretable representation of the audio signal [2]. Recently, the application of learnable time-frequency transformations has emerged as a popular alternative [6]. These methods formulate the time-frequency transformation as a component of the network defined by a set of parameters $\theta$. These parameters can then be trained, allowing the network to learn an optimal transformation [6]. We provide more detail on learnable time-frequency transformations in Section 3.2.2.

Once a time-frequency representation of the audio signal has been produced, a CNN is employed, extracting relevant features from the spectrograms [29][2][6]. This is generally formulated as a classification problem, where the CNN is trained to identify important patterns within the spectrogram that are useful for discerning the class each audio signal belongs to [6]. By doing so, the CNN learns meaningful representations that can be used to discriminate between different types of audio samples.

For classification purposes CNNs are structured such that the final layer contains $m$ nodes, where $m$ is equal to the number of classes in the classification problem [6]. This is often quite a small number, resulting in a limited capacity to represent information. Moreover, the values in the final layer are fine-tuned to describe how likely an input is to belong to each given class, rather than describing the input itself. As such, researchers opt to take the output of the penultimate fully connected layer as the audio embedding [6]. This layer synthesizes information from the convolutional operations into a vector of arbitrary dimensionality. Doing so results in a more generalizable embedding, better suited to describing the contents of the audio signal.

## 2.4 Performance metrics

### 2.4.1 Inception score (IS)

The Inception score (IS) [23] incorporates a pre-trained Inception-v3 model [30] to measure the quality of generated images. To do so, generated images are passed to the Inception-v3 model to ascertain a predicted class distribution $p(y|\mathbf{x})$. If the generator produces images representative of a particular class, the conditional label distribution $p(y|\mathbf{x})$ should have relatively low entropy [23]. In addition, to evaluate the variety of generated images, the marginal distribution $\int p(y|\mathbf{x} = G(z))dz$ is calculated, where $G(z)$ represents the generator function [23]. If the marginal distribution has high entropy, $G(z)$ can be understood as producing diverse images. The IS quantifies both criteria by employing the following equation:

$$IS = \exp(\mathbb{E}_x KL(p(y|\mathbf{x})||p(y)). \tag{2.6}$$

This provides a measure of the average divergence between the marginal and conditional distribution over all generated images. Given that we have opposing objectives for both $p(y)$ and $p(y|\mathbf{x})$, we desire the two distributions to be farther apart. Consequently, larger values of IS are desirable. The main limitation of the IS is that it does not compare generated images to real images, thus it can be fooled by unrealistic images that the Inception-v3 model is confident about [31].

### 2.4.2 Fréchet Inception Distance (FID)

As with the IS, the Fréchet Inception Distance (FID) also utilizes the Inception-v3 model to evaluate performance of image generators [31]. The distinction lies in how the Inception-v3 model is incorporated. Rather than evaluating the output distribution, FID measures the distance between feature representations of real and generated images [31]. Specifically, the mean and covariance of node activations for real $(\mu_r, \Sigma_r)$ and generated $(\mu_g, \Sigma_g)$ images are

taken from the final layer of the Inception-v3 model [31]. It is assumed that these activations come from a multidimensional Gaussian [31], thus the distance between the two distributions is given by the equation below, where *Tr* denotes the trace.

$$FID = \|\mu_r - \mu_g\|_2^2 + Tr\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right) \tag{2.7}$$

The first component $\|\mu_r - \mu_g\|_2^2$ captures the $L_2$ distance between means for real and generated images, while the second $Tr\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right)$ measures the total variance between the distributions. Smaller FID scores indicate greater similarity between real and generated images and vice versa. Consequently, smaller FID scores indicate better performance.

Importantly, FID addresses the major limitation of the IS by comparing generated images to real ones. This makes the measure harder to fool, and has been shown to better align with human judgement [31][32]. As such it is generally regarded as a better measure of image quality [31].

### 2.4.3 Image Classifier Networks

Image classifiers are commonly applied in simple Audio-to-Image generation problems to quantify conditional adherence [2][3][29]. By conditional adherence we refer to the extent to which the generative network produces samples that align with conditioning information.

To measure this, researchers train and evaluate an image classifier (typically a CNN) on real images from a given dataset. Importantly, this dataset is the same one used to train the generative model [2]. By doing so, the classifier develops a capacity to identify the contents generated images. The classifier can then be applied to a set of generated images where the conditioning information is known. Researchers then measure the extent to which classifier predictions align with conditioning information to calculate a measure of conditional adherence [2].

It is important to note that this method can only be reliably implemented when the training dataset is easy to classify. This is because the image classification network must achieve near perfect accuracy to give a reliable performance measure on generated samples.

## 2.5 Audio-to-Image

In the following section, we shift focus to discussing existing literature within the Audio-to-Image generation domain. We highlight the key contributions that have significantly impacted the field, with specific focus on those that are most relevant to the current research project. Due to constraints defined by the project's scope, this review will not encompass all aspects of the domain. This said, the works included have been carefully curated to ensure the review provides a sufficient amount of detail into the most pertinent aspects of the field.

### 2.5.1 Deep Cross Modal Audio Visual Generation

In 2017, Chen et al. [2] published the seminal Audio-to-Image generation paper. The authors used a GAN, conditioned on musical audio samples to synthesize $64 \times 64 \times 3$ images of individuals playing musical instruments [2].

Their generative framework was comprised of two separate models. An audio encoding network was first used to process a time-frequency transformation of the raw audio signal using a CNN [2]. The authors considered various time-frequency transformations, comparing model performance with each. Specifically, they assessed the classification accuracy of

the audio encoding model using five different transformations. Those being: Short-Time Fourier Transform (STFT), Constant-Q Transform (CQT), Mel-Frequency Cepstral Coefficients (MFCC), Mel-Spectrum (MS) and Log-Mel Spectrum (LMS) time-frequency transformations [2]. LMS transformations were found to perform the best (see Table 2.2). Consequently, the authors opted to use these as input to their embedding network. A visual comparison of different time-frequency transformations can be found in Figure 2.2. In this, we see that LMS outputs appear to contain the most information about the audio signal. This additional information could explain the improvement in performance.

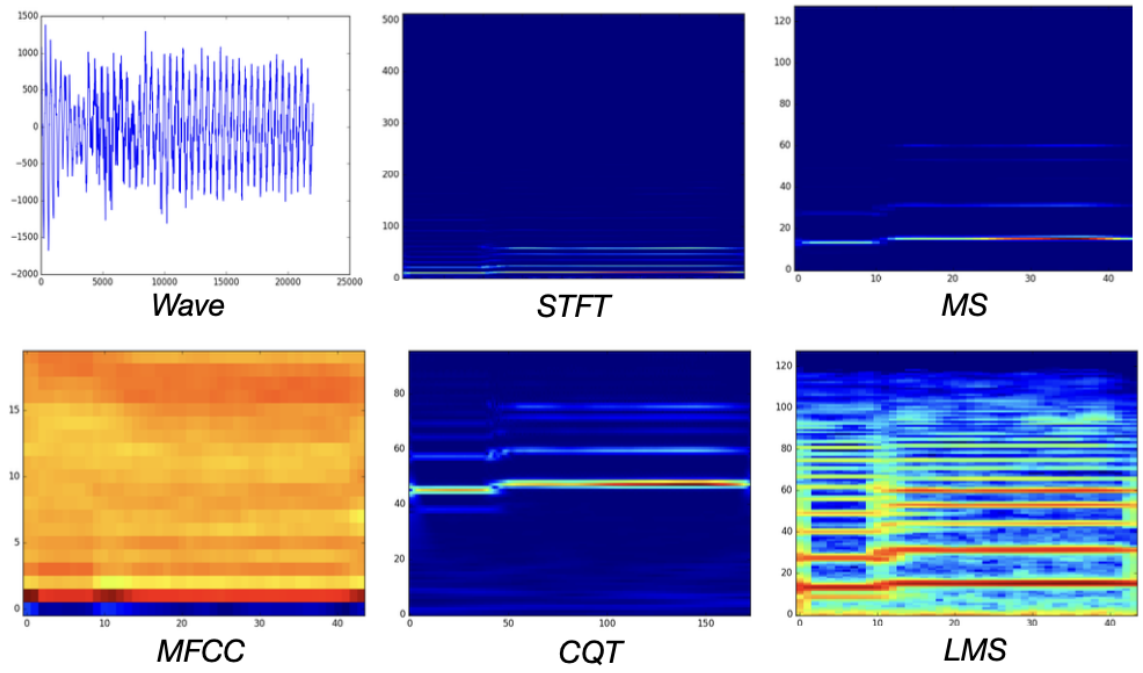



Figure 2.2: Visualization of different time-frequency transformations. Image taken from [2].

The authors passed embeddings obtained from the encoding model to the generative network, where their dimensionality is reduced via a leaky ReLU activated fully connected layer. This operation speeds up training, and also provides the generative network the opportunity to determine which aspects of the embedding are most salient to the image generation process [2]. The processed embedding is then concatenated with random noise and passed through a series of convolutional up-sampling blocks to produce a generated image [2]. A discriminator is employed to determine the origin of the generated sample. As discussed in Section 2.3.1, both networks are trained adversarially, allowing the generator to implicitly learn the underlying data distribution.

A separate classification network was incorporated to determine whether generated images corresponded to the correct class. Findings indicated that samples aligned with conditioning information 75.56% of the time [2]. The authors refer to their Audio-to-Image generation model as S2I-C [2].

### 2.5.2 CMCGAN

The authors of CMCGAN [3] incorporate principals derived from CycleGAN [33] to create a novel end-to-end Audio-to-Image generation network. Moreover, they do so while also increasing image resolution to $128 \times 128 \times 3$ [3]. We use end-to-end in the typical sense,

Table 2.2: Table of audio encoder classification accuracies across five different time-frequency transformations [2].

| | **MS** | **LMS** | **CQT** | **MFCC** | **STFT** |
|---|---|---|---|---|---|
| Classification Accuracy | 66.09% | **87.44%** | 77.78% | 81.05% | 75.73% |

describing a network whereby all sub-components (e.g. audio embedding network and image generation network) are trained concurrently.

The proposed model is comprised of two encoders, one for audio data (EA) and one for image data (EI), and two decoders for the same modalities (DA and DI). EA and EI are responsible for generating latent data representations from their inputs, those being log-mel spectrograms and images of individuals playing musical instruments respectively [3]. The decoders take these latent representations as input and convert them into log-mel spectrograms (DA) and images (DI). Importantly, decoders are capable of taking inputs from the output of either encoder [3]. A helpful illustration of the architecture employed can be found in Figure 2.3.

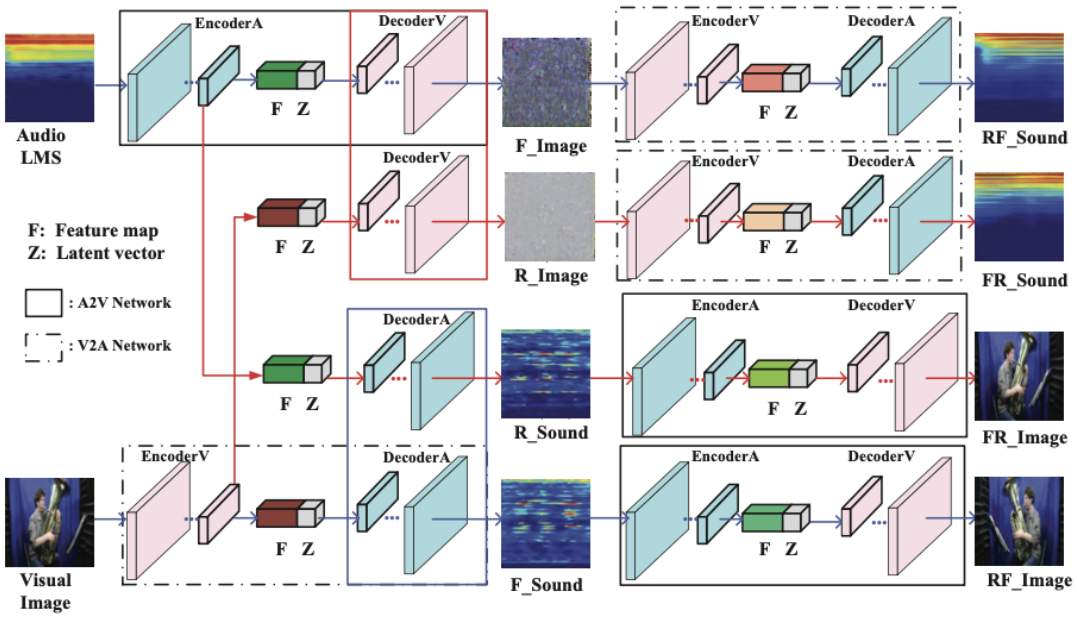



Figure 2.3: CMCGAN architecture. Image taken from [3].

During the training process, both decoders receive inputs from either encoder and generate outputs corresponding to their respective modalities (i.e. images for DI and spectrograms for DA). A discriminator is then employed to determine whether decoder outputs originated from latents produced by encoders of the same modality (positive instance) or alternate modality (negative instance) [3]. By optimizing both encoders and decoders to fool this discriminator, decoder networks learn to produce consistent outputs for both input modalities [3]. Importantly, this does not imply that the output must resemble the input from the opposite modality, only that outputs ought to bear significant similarities to one another.

The network also incorporates *cycle consistency loss*, the inclusion of which is necessary

for the generation of meaningful cross-modal outputs. This is achieved by arranging the encoders and decoders to convert an input (audio/image) into an intermediate representation of the other modality. This intermediate representation is then converted back into the original modality. The loss measures the discrepancy between the original and reconstructed output. In order to minimize this loss, the intermediate representation must retain a sufficient amount of information about the original input. Given that the network is trained to produce comparable outputs regardless of input modality, once trained, the network can be arranged so that it is able to convert inputs from one modality into the other [3].

In the context of Audio-to-Image conversion, audio encoders and decoders are used to generate an intermediate audio representation. Subsequently, the audio encoder and image decoder are used to convert this into an instance of the opposite modality [3]. We refer readers to Figure 2.3 for a visual overview of this process.

This methodological approach allows for the joint training of both audio encoding and image generation networks [3]. Doing so allows the networks to provide feedback to one another during training, and potentially facilitates the learning of more complex relationships that would not have been learnt if the networks were trained in isolation [3]. Results revealed a modest improvement in adherence to conditioning information when compared to S2I-C, with a classification accuracy of 76.61% on examples withheld during training [3]. Moreover, a notable improvement to FID and IS scores was also observed. We refer readers to Table 2.3 for the specific numbers.

### 2.5.3 Cross Modal Contrastive Representation Learning

In their 2022 work, Chung et al. [4] proposed two significant contributions to the Audio-to-Image domain. Firstly, they integrated a contrastive loss function into the audio encoder's training, encouraging greater separation between embeddings of different audio classes. Secondly, they incorporated a self-attention module into their generative network to capture long-range dependencies within images [4]. Notably, their work also increased image resolution to $256 \times 256 \times 3$ [4].

The contrastive loss function operates on embeddings from both an audio and visual encoding network, and serves two purposes. Firstly, it encourages similarity between embeddings originating from audio samples of the same category, while promoting dissimilarity among those from different categories [4]. It is reasoned that by doing so, it becomes easier for the generative network to distinguish different conditioning instances, thus helping it produce correctly aligned samples [4]. Secondly, the loss function also pushes associated audio and image embeddings closer together in space (and vice versa for non-associated embeddings) [4]. The authors argue that doing so helps encoded visually meaningful information into audio embeddings. A technique that has been associated with improved visual quality in generated samples [14]. A visualization of the training procedure can be seen in Figure 2.4.

Table 2.3: Image evaluation comparison between state-of-the-art GAN Audio-to-Image generation networks [4].

| **Method** | **IS** | **FID** |
|---|---|---|
| S2I-C | 2.315 | 252.66 |
| CMCGAN | 2.883 | 215.92 |
| Classification Learning + SAGAN | 2.757 | 178.89 |
| CMCRL + SAGAN | 5.288 | 107.26 |

The network's self-attention component draws inspiration from the architecture of SAGAN [34]. SAGAN incorporates a self-attention function between convolutions to encourage better long-range dependencies within the output. By long-range dependencies, we refer to visual relationships in images that are separated spatially. For instance the relationship between a shadow and light source. These dependencies play a crucial role in building a cohesive and meaningful output [35], and ought to be accurately captured.

Results showed that by incorporating the above techniques, the authors were able to produce conditionally aligned images 89.07% of the time. This is a substantial improvement from previous works on the same dataset [2][4]. Moreover, generated images displayed better FID and IS performance than previous models as well, indicating an improvement in visual quality. We refer readers to Table 2.3 for the exact results

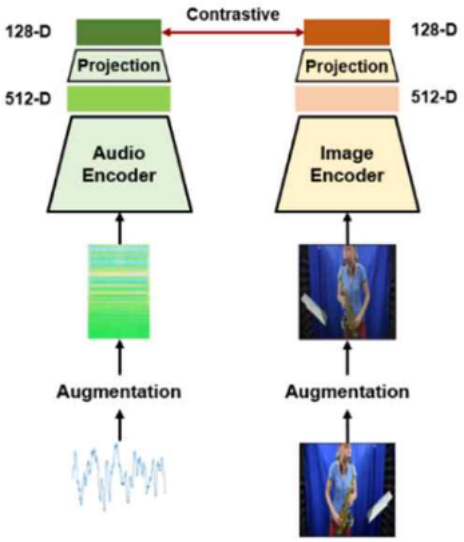



Figure 2.4: Visualization of the cross-modal contrastive loss training procedure. Image taken from [4].

### 2.5.4 AudioCLIP

Guzhov et al. [6] recently proposed AudioCLIP, extending the highly expressive Contrastive Language Image Pretraining (CLIP) [17] network to the audio domain.

CLIP is an encoding model comprised of both a text and image sub-network [17]. The sub-networks were jointly trained on an extensive dataset of over 400-million image-text pairs [17], and are responsible for encoding instances from their respective modalities into embedding vectors. A contrastive loss was applied between sub-networks encouraging the maximization of cosine similarity between paired text and image embeddings, and minimization of similarity between disparate instances [17]. As with Chung et al. [4], this fostered a strong separation between different groups of embeddings, and a large degree of similarity between modalities [17].

CLIP's greatest advantage stems from the scale of its training dataset, allowing the model to learn highly nuanced and intricate patterns, enhancing the representative power of its embeddings [17]. Ramesh et al. [14] found these qualities to benefit image generation significantly in their Text-to-Image model DALL-E 2. We discus this in more detail in Section 2.6.2.

Guzhov et al. [6] augment CLIP, introducing an audio encoding sub-network into the model. They opted to use a pre-trained ESResNe(X)t-fbsp network to achieve this due to its state-of-the-art performance on two major audio classification datasets (UrbanSound8K and ESC-50 [36][6]). Notably, ESResNe(X)t-fbsp forgoes the use of log-mel spectrograms as inputs to the model and instead incorporates trainable time-frequency transformations due to their increased flexibility [36]. We provide more detail on this in Section 3.2.2. The author's then used paired audio-image and audio-text datasets, alongside a contrastive loss to align the audio-head with CLIP's text-and-image-heads.

Results showed that the audio-head achieved new state-of-the-art performance on both ESC-50 and UrbanSound8K [6], improving upon the baseline ESResNe(X)t-fbsp network [36]. While CLIP proved highly effective for Text-to-Image generation, and strong evidence exists indicating the effectiveness of contrastively trained embedding networks for Audio-to-Image generation [4], no one to date has incorporated AudioCLIP's audio-head into an Audio-to-Image generation framework.

### 2.5.5 Audio-to-Image Summary

While this synopsis does not encompass every contribution made within the Audio-to-Image generation domain, it does offer a curated subset of relevant information. We showed how Chen et al.'s inaugural work [2] demonstrated the feasibility of the problem, and highlighted how log-mel spectrograms could be used to produce conditioning embeddings. We also showed how CMCGAN's effective integration of an end-to-end learning approach was able to enhance conditional adherence and image quality [3]. Building upon this development, we then highlighted how Chung et al. [4] illustrated that further improvements could be realized via the incorporation of attention calculations during image generation and contrastively trained embedding networks. Lastly, we noted the relevance of Guzhov et al.'s [6] work extending CLIP [17] to the audio domain, and cited the potential benefit it could provide if incorporated into an Audio-to-Image generative framework.

Whilst previous works managed to increase image resolution to $256 \times 256 \times 3$, the size of the generated samples remains modest. In addition, conditional adherence and FID scores leave significant room for improvement. With only 89% of generated samples aligning with conditioning information in the best approach [4], it is difficult to argue that current methods have reached an acceptable level of reliability. Furthermore, FID scores below 15 are commonly achieved by state-of-the-art Text-to-Image generation models [14][13], highlighting a noticeable gap in image fidelity.

Finally, all of the studies discussed employed the SubURMP dataset to the generative problem. This is understandable given the limited amount of publicly available datasets. However, the dataset lacks diversity and is restricted to audio and image samples of musical instruments [2]. Therefore, it would be interesting to evaluate performance of these models on larger scale datasets. We note that other efforts have been made in the space to achieve this [11], although none have been as large scale as recent Text-to-Image generation models [12][13][14].

## 2.6 Text-to-Image

In the following section we highlight several recent innovations within the related field of Text-to-Image generation. Although not the primary focus of this research project, the domain has the potential to offer insights that could be effectively integrated to into Audio-to-Image generative problems. In light of this, we provide a high-level review of Text-to-Image

generation models, placing particular emphasis on those contributions that could benefit the Audio-to-Image generation domain.

### 2.6.1 GLIDE

GLIDE [15] was one of the first works to incorporate a diffusion model into a Text-to-Image generative framework. The network incorporates a large transformer model [18] to encode text into embeddings. Transformer models stack layers of attention calculations on top of each other, iteratively learning relationships between word tokens [18], and have proven to be highly effective solutions to natural language processing tasks [37][38]. We refer readers to the work of Vaswani et al. [18] for more detail.

Whilst the encoder provided rich textual embeddings, encouraging the diffusion model to produce samples that aligned with this information proved challenging [15]. The authors addressed this by employing several conditioning strategies into the diffusion network. Firstly, they added text and time-step embeddings together, using the combined embedding as input to the network. Recall that diffusion models require knowledge of the time-step as the amount of noise added to an image is dependent upon this information (see Section 2.3.3). Whilst this helped, it proved insufficient on its own [15]. To further encourage alignment between conditions and generated samples, the authors also concatenated text embeddings with image features during U-Net attention calculations [15]. This effectively creates a cross-attention mechanism, allowing the network to continually adjust its feature representations to align with text conditions. Combining both approaches was found to help guide the generative process in the direction of conditioning information [15].

In addition, the authors also experimented with both classifier and classifier-free guidance, to further guide the generative process in line with conditioning information [15]. Human evaluations, the gold standard image generation metric, showed classifier-free guidance to be more effective at producing conditionally aligned samples [15]. Moreover, when comparing samples produced by GLIDE to GAN alternatives, they found that their approach significantly improved FID scores [15].

### 2.6.2 DALL-E 2

Ramesh et al. [14] build upon GLIDE [15], providing three primary contributions to the domain. Firstly they use CLIP's text-head as their encoding network [14][17]. As discussed in Section 2.5.4, CLIP's contrastive training procedure, coupled with its sizable training dataset, facilitates the generation of visually meaningful and highly discriminable text-embeddings [17][14].

The author's second contribution pertains to the inclusion of an additional diffusion network that they refer to as the prior. The prior serves the role of mapping CLIP audio embeddings into the image embedding space [14]. Whilst CLIP's training procedure encourages alignment between modalities, the authors argue that they still occupy disjoint sets [14]. By incorporating a prior network, they reason that more visually meaningful information will be incorporated into embeddings, improving generative performance [14].

Their third contribution regards image resolution. Previous Text-to-Image models had struggled to generate high resolution outputs, with GAN approaches not exceeding $256 \times 256 \times 3$ [39]. These challenges were attributed to a lack of overlapping supports between image scales [40], which coupled with training instability, made high resolution image generation with GANs infeasible. Ramesh et al. [14] were circumvented these challenges by incorporating multiple diffusion super-resolution networks. These were found to be stable and scalable solutions to the problem, facilitating the generation of $1024 \times 1024 \times 3$ images.

DALL-E 2's outputs were more diverse and achieved better FID scores than GLIDE [15], although were found to be less semantically consistent [14]. We reason that this discrepancy is likely a result of the diffusion prior for two reasons. Firstly, the randomly sampled noise that initializes the reverse process will introduce additional variation into embeddings. It is reasonable to believe that this variation could produce more diverse samples, while simultaneously shifting conditions away from their original meaning, reducing semantic consistency. Secondly, if the image embedding space contains more visually meaningful information, it is reasonable to believe that the text embedding space may contain more textually relevant information. If so, mapping text conditions into the image space would produce more visually meaningful but less textually relevant conditions. Logically, this would be associated with improved FID scores (image quality) and outputs that were less aligned with textual information.

### 2.6.3 Latent Diffusion Models

In their paper, Rombach et al. [12] investigate the potential computational savings that latent diffusion models could provide. They accomplish this using an auto-encoder to project images into a lower-dimensional latent space [12]. The network is trained such that its decoder is able to reproduce images from latent representations while retaining as much information about the original image as possible [12]. They then train a diffusion model on latent image representations and use the the decoder to convert its outputs into visually recognizable images [12].

Their approach yields significant computational benefits, while largely retaining image quality. Specifically Rombach et al.'s [12] model contained 1.4B parameters, as opposed to GLIDE's 6B [15]. Moreover, the authors provide theoretical justification for the appropriateness of working with latent representations. They argue that images often contain high-frequency details that are imperceptible to humans. Likelihood functions are prone to over-prioritizing this information, leading to networks that allocate significant resources towards perceptually irrelevant details [12]. By incorporating an auto-encoder, Rombach et al. remove a portion of these high frequency details, thus saving computational resources without noticeably affecting image quality [12].

Building upon the foundations laid by GLIDE [15], Rombach et al. [12] also investigate alternative mechanisms for incorporating conditioning information into the architecture of the U-Net. Specifically, rather than concatenating text embeddings with image features in attention calculations, Rombach et al. opt to incorporate a stand alone cross-attention calculation. In this configuration, image segments are projected into the query vector while text conditions are projected into key and value vectors [12].

Doing so offers several advantages when compared to GLIDE's combined calculation [15]. By conducting a specific cross-attention operations the model can concentrate directly on the alignment between image features and textual information. In contrast, GLIDE's combined approach could make it more difficult to learn these relationships as the network is simultaneously attempting to learn relationships between image components. It is worth noting that learning relationships between image components is also highly important. As a result, Rombach et al. [12] also integrate a self-attention mechanism between image features into the network.

## 2.7 Summary

In this Chapter, we provided a comprehensive overview of work relevant to the current research project. We began with a thorough explanation of the foundational concepts neces-

sary to engage with this work, then transitioned into a discussion of relevant model architectures. Important performance measures were discussed prior to detailing recent works within the Audio-to-Image generation domain. Finally state-of-the-art Text-to-Image generation models were discussed.

In addition to this, we also highlighted several limitations present within the Audio-to-Image generation domain. These include small image resolutions, lack of conditional adherence and image fidelity, alongside the simplistic nature of the SubURMP dataset.

Similar issues with image resolution were observed within the Text-to-Image generation space [40]. None the less, these were largely mitigated when diffusion models replaced GANs as the primary generative framework [14][13]. One reason for this can be found in our discussion on model architecture. Here we noted that GANs can be challenging to train at scale, due to their implicit likelihood function. In contrast, diffusion models provide a tractable explicit likelihood objective, lending itself to more efficient scaling [27].

The scalable nature of diffusion models also makes them easier to apply to larger datasets. Moreover, diffusion models have been observed to achieve significantly better FID scores than GANs in the Text-to-Image generation domain [14][13]. This makes them an interesting option when considering strategies to improve the aforementioned Audio-to-Image generation limitations.

Finally, with respect to improving conditional adherence, we noted that the integration of a cross modal contrastive loss during training led to performance improvements in Chung et al.'s [4] work. We wonder whether additional performance improvements could be realised if a more expressive cross modal, contrastively optimized encoding model such as AudioCLIP, is incorporated.

# Chapter 3

# Model Architecture, Design and Justification

## 3.1 Chapter Overview

The this chapter we provide a detailed breakdown of our proposed Audio-to-Image generation model. We highlight the design and provide justification for our architectural decisions. By doing so, it is our intention that readers will gain a comprehensive understanding of the architectural contributions of this research project.

We break this chapter into two primary sections, each dedicated to an individual sub-network of the model. In the first section we detail the audio encoding network, responsible for converting audio samples into embeddings. The second section describes the diffusion network, that uses these embeddings to condition the image generation process.

Both sections are comprised of three sub-sections. The first provides readers with relevant background information. This is followed by a discussion of the architecture employed, where we outline pertinent technical aspects of the sub-network. Lastly, we show how each sub-network is incorporated into the broader Audio-to-Image generative framework. A diagram of the overall model can be found below in Figure 3.1.

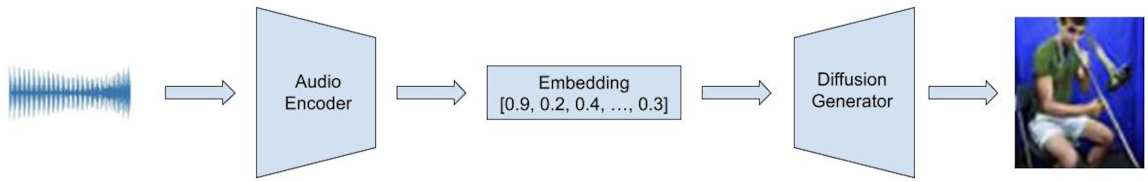



Figure 3.1: Diagram of the proposed Audio-to-Image generation network architecture.

## 3.2 Encoding Model

### 3.2.1 Background

We use AudioCLIP [6], and more specifically its audio-head, as the encoding network in our conditional generative framework. We opted to do so for several reasons. Firstly, the successful application of CLIP [17] in DALL-E 2 [14] provides an indicative example of the potential benefit the network could bring. Moreover, AudioCLIP's audio-head has demonstrated state-of-the-art performance on challenging classification tasks [6], further supporting its potential utility. Finally, given that the audio-head is contrastively trained alongside

CLIP's highly expressive image encoding network, the embeddings produced ought to contain visually meaningful information [6]. The inclusion of which has been associated with improvements in the visual quality of generated images [22][41][14]. Despite this, no work has yet investigated the potential benefits the incorporation of AudioCLIP could bring the Audio-to-Image generation domain.

We do note that DALL-E 2 [14] generated less semantically consistent images than the GLIDE [15]. It is important to acknowledge that this performance discrepancy can reasonably be attributed to DALL-E 2's integration of a diffusion prior, rather than a limitation of CLIP itself. We refer readers back to Section 2.6.2 for more detail.

### 3.2.2 Architecture

In this sub-section, we provide an overview of the relevant architecture incorporated by AudioCLIP. We do so in more detail than was provided in our literature review given the different focus of the current section. That being, to provide readers with a *detailed, and technical* understanding of the model such that they are well-positioned to engage with future aspects of this research project.

As mentioned previously, AudioCLIP is comprised of three sub-networks, those being its text, image, and audio-heads [6]. While text and image-heads do not produce audio embeddings, they serve an important role in the contrastive training procedure. Hence, we provide a high level description of both sub-networks below.

The text-head is a modified transformer [18], consisting of 12 layers of stacked attention, each layer containing 8 attention heads [17]. The image-head employs a ResNet-50 as its architectural backbone, and was chosen given its proven capacity to generate powerful image representations [17].

The most relevant sub-network to the current problem is the audio-head. Employing a ESResNe(X)t-fbsp model [36], the audio-head was original proposed for environmental sound classification, where it achieved state-of-the-art performance on numerous benchmarks [36]. The networks design is predicated upon the ESResNet [5], an illustration of which can be found in Figure 3.2.

ESResNet applies convolutions (red blocks) followed by a batch normalization layer (green blocks) to a Short-Time Fourier Transform (STFT) spectrogram [5]. The resulting information is then passed through four residual layers (blue blocks). Each residual block consists of a stack of three bottleneck layers (yellow blocks), containing multiple 1D convolutions and batch normalization operations [5]. The output of each residual block is combined with an attention-augmented input (violet blocks). Finally, an average pooling layer (grey block) is applied, and the output is passed to a fully connected layer (black block) for classification [5].

The ESResNeXt network improves this design via the integration of cardinality [36]. Originally proposed by Xie et al. [42] cardinality involves stacking several smaller residual channels on-top of one another and passing a different portion of the input to each. This strategy allows each channel to *focus* on a distinct aspect of the input, decentralizing the learning process and avoiding the potential bottleneck that could occur with one large channel [42].

The ESResNe(X)t-fbsp model improves this once again by incorporating learnable time-frequency transformations [6]. Rather than converting audio samples into log-mel spectrograms, the network creates its own inputs via a parameterized kernel [6][36]. These transformations are founded upon complex B-spline wavelets [43]. The kernel defining these wavelets is presented in Equation 3.2, where $m$ represents the order, $f_b$ signifies the bandwidth, and $f_c$ denotes the central frequency. In the trainable formulation, the parameters $m$,

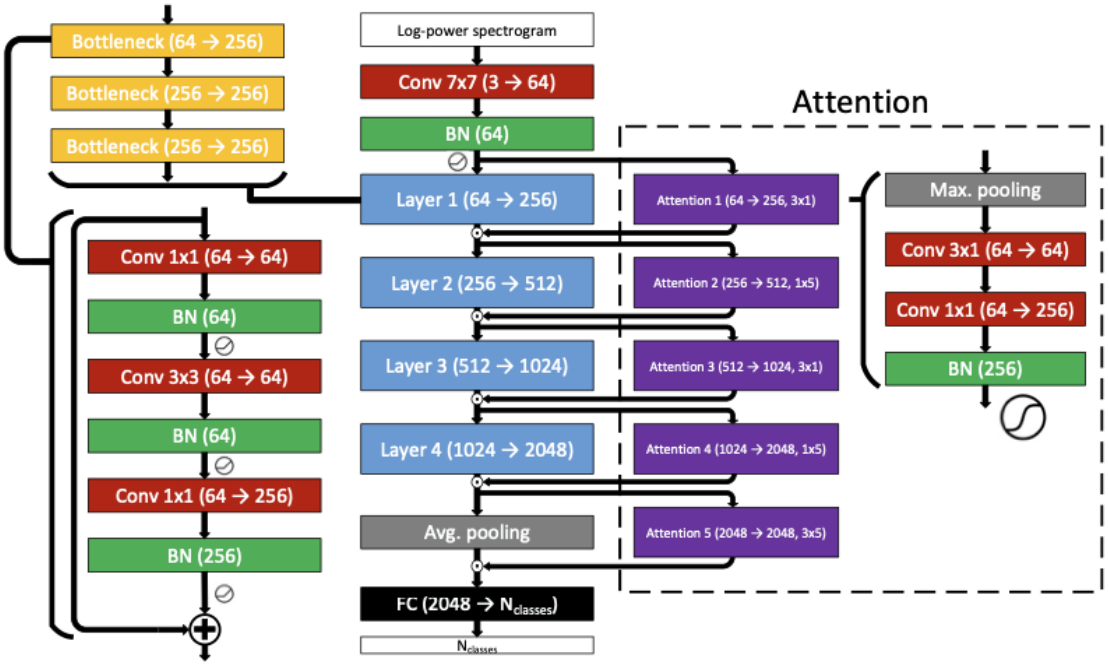



Figure 3.2: ESResNet architecture. Image taken from [5].

$f_b$, and $f_c$ are associated with learnable weights, determining their respective contributions to the output [36]. As these weights change, the time-frequency transformation does too. This allows the network to learn a transformation of the waveform it deems most effective. Doing so has been found to result in better performance than fixed transformations [36].

$$K_{DFT}^{f_c}(n) = e^{-2i\pi f_c n} \tag{3.1}$$

$$K_{fbsp}^{m-f_b-f_c}(n) = \sqrt{f_b}\left(\frac{f_b n}{m}\right)^m e^{2i\pi f_c n} \tag{3.2}$$

To better understand why this particular kernel is used, we draw comparison to the Discrete Fourier Transform kernel employed in the Short-Time Fourier Transform (STFT). The formula for this kernel is presented in Equation 3.1. By setting $m = 0$ and $f_b = 1$ in Equation 3.2, each kernel can be understood as the inverse of the other [36]. Given that STFT are common time-frequency transformations, this parameterization provides a useful starting point for training. The authors employ this initialization, allowing the network to iteratively improve upon a known and effective time-frequency transformation [36].

### 3.2.3 Implementation

Two pre-trained versions of AudioCLIP are made publicly available on GitHub [44]. We opt to utilize the *partially trained* weights rather than the fully trained alternative. The difference being that the partially trained model freezes the weights of the original CLIP text-and-image-heads during training [6]. Doing so ensures their highly expressive capabilities are retained as the model learns, avoiding issues associated with catastrophic forgetting [6][45].

The audio-head is initially pre-trained on an audio classification task, laying the groundwork for its integration into the AudioCLIP framework [6]. The network's classification

head (final layer) is removed, and the penultimate fully connected layer is used to produce the embedding [6]. This layer contains 1024 output nodes, producing 1024-dimensional embeddings, aligning it with the dimensionality of CLIP text-and-image-heads [6][17]. The audio-head is then contrastively trained against frozen text-and-image heads shifting audio embedding to align with CLIP's latent space [6]. Paired audio-image and audio-text datasets are used to facilitate this [6]. A visual aid illustrating the contrastive training procedure can be found in Figure 3.3.

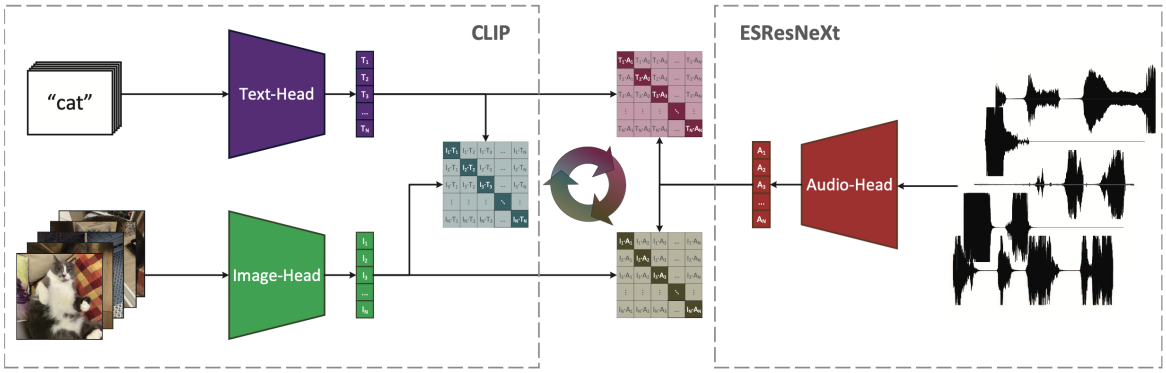



Figure 3.3: Illustration of the contrastive training procedure employed by AudioCLIP. Image taken from [6].

We incorporate this version of the audio-head as the encoding network in our baseline framework. We refine the network for the current task by conducting several experiments upon it. These are outlined in Section 4.3.1. Ultimately, AudioCLIP's audio-head serves the role of producing the audio embeddings which condition the image generation process.

## 3.3 Generative Model

### 3.3.1 Background

Previous Audio-to-Image generation research has primarily employed GANs to synthesize images [2][4][3]. At the time of writing, no studies have investigated the potential benefits that diffusion models may provide to the generation problem. We deem this to be an interesting avenue of exploration for a variety of reasons.

One motivation arises form the successful application of diffusion models on Text-to-Image generation problems [15][14][12]. While only observational, the similarities between the domains provide good reason to believe that diffusion models could also be incorporated into Audio-to-Image generation frameworks. If so, it may be possible to realise some of the same performance improvements diffusion models have provided to Text-to-Image synthesis tasks. In addition, there are also a number of theoretical justifications making diffusion models an appealing alternative to GANs.

Firstly, GANs can prove highly challenging to train, requiring careful and considered hyper-paramater tuning to achieve satisfactory results [46]. Moreover, GANs are known to experience mode collapse [27][47] whereby they only learn to generate a limited subset of all possible outputs. This translates to a lack of diversity in generated images. In contrast, diffusion models have been shown not to suffer from the same limitations [27]. This is partially due to differences in their optimization criteria [46].

GANs have a non-static optimization criteria, taking the form of a two player adversarial game upon the value function described in Section 2.3.1. As both the generator and discrim-

inator learn, the optimization criteria changes. This inherent fluidity in model focus has the potential to make it challenging to learn the desired data distribution. In contrast, diffusion models have a static likelihood objective over the data distribution avoiding this issue [27]. Moreover, they have been shown to generate more diverse samples, addressing issues associated with mode collapse, and have proven to be highly scalable with predictable training behaviour [27][14][13].

These reasons have guided our decision to investigate whether a diffusion model can be effectively incorporated into an Audio-to-Image generation framework. This said, it is important to note that diffusion models are not a panacea to the generative task. They can be highly computationally expensive and slow to inference from due to the sequential nature of the reverse noising process [26]. This is a serious limitation as this process tends to contain somewhere in the region of 1000 steps [12][13][14][15]. In addition, diffusion models can struggle to adhere to conditioning information, often requiring a variety of different techniques to overcome this issue [12][13][14][15][28]. Nonetheless, investigating how they can be applied to Audio-to-Image generation problems remains an interesting research question.

### 3.3.2 Architecture

In the following section, we present a detailed overview of the architecture used to construct our baseline diffusion model. Our network is designed on top an open sourced *class conditional* diffusion model developed by Capelle [48]. We adapt Capelle's program, tailoring it to the Audio-to-Image generation task.

The network is comprised of several different computational components. The first two of which are responsible for processing conditioning information, while the latter four comprise the U-Net. We outline each component below, then show how they are arranged within the network.

#### Encoding Layer

The encoding layer is the first component of the network. It takes a 1024-dimensional audio embedding as input, and projects it into a lower 256-dimensional representation via a leaky ReLU activated fully connected layer. We opted to follow this approach as it is in line with Chen et al.'s [2] previous work discussed in Section 2.5.1.

The incorporation of this layer reduces computational complexity via the associated dimensionality reduction, and allows the network to *learn* which components of the audio embedding are most salient for image generation [2].

#### Positional Encoding

A positional encoding module follows the encoding layer, and is responsible for the creation of time-step embeddings. These provide critical information to the diffusion model as the amount of noise contained within an image is a function of the time-step $t$ (refer to Section 2.3.3 for more detail). We opt to use sinusoidal embeddings to encode time-step information. Proposed by Vaswani et al. [18], sinusoidal embeddings use sine and cosine functions to create unique, smooth, and continuous encoding vectors for each time-step [18]. Moreover, the approach is purely mathematical, making it highly efficient to integrate into the network [18].

### Down-Sample Blocks

Down-sample blocks are the first component of the U-Net. These reduce the size of an input, and extract meaningful patterns from within them. Each block begins with a max pooling kernel, shrinking the dimensionality of the input. By incorporating this function at the start of the block we improve computationally efficiency, as subsequent operations are performed upon a smaller input [13].

Following this, two convolutional operations, separated by group normalization layers and GeLU activations are applied. This aligns with common approaches found within the literature [12][48]. These operations are responsible for converting the input into meaningful feature representations. For ease of reference, we refer to this sequence of operations as a *DoubleConv*.

### Up-sample blocks

Up-sample blocks perform the opposite function, increasing input resolution. To achieve this they use bilinear interpolations to infer intermediate pixel values based upon the values of surrounding pixels.

A skip connection is also applied between down-and-up-sample blocks of the same resolution. This is done by concatenating down-sample outputs to the output of the up-sampling operation. By doing so, we retain information from the down-sample block that may have been lost during further processing [7]. A DoubleConv block is then incorporated to extract patterns from the up-sampled features.

### Cross-Attention Blocks

We incorporate cross-attention blocks in accordance with the implementation proposed by Rombach et al. [12]. We first compute self-attention between image components, allowing the network to learn the global relationships present in within the image features. Cross-attention is then calculated between image features and audio conditions, encouraging alignment between the two [12].

### Bottleneck Layer

The final computational component of the U-Net, in accordance with standard practice, is the bottleneck layer [26][15][14][12][13]. Here, two sets of DoubleConv's are applied to the lowest dimensional representation of the input, further processing it prior to up-sampling.

### U-Net

The U-Net is formed by combining the four aforementioned computational blocks. A stack of three down-sample blocks are used to compresses and transform the original input into a low-dimensional feature representation. This is passed through the bottleneck layer and then a stack of three up-sample blocks, producing an output of the same shape as the original input. Each up-and-down sample block within their respective stacks are separated by a cross-attention block, continually aligning image features with audio conditions.

### Overall Structure

The entire network is formed by combining the U-Net and embedding layers. Audio embeddings are first compressed into a 256-dimensional representation via the encoding layer

and 256-dimensional time-step embeddings are computed via the positional encoding block. These are then added together, combining their information in line with the approach outlined in GLIDE [15]. This information is then passed to the U-Net alongside an input image with $t$ steps of noise added to it. The U-Net uses this information to produce a feature representation of the input containing $t-1$ steps of noise. We note that the combined conditioning information is integrated multiple times throughout the U-Net, specifically by adding it to to output of each up-and-down-sample block. The U-Net's output is then transformed via a single convolutional operation, producing a $64 \times 64 \times 3$ RGB image. This structure follows standard practice employed by existing Text-to-Image conditional diffusion models [12][13][14].

### 3.3.3 Implementation

The diffusion model described above is trained in accordance with the algorithm detailed in Ho et al.'s paper, Denoising Diffusion Probabilistic Models [7]. A description of this procedure is provided below, alongside pseudo-code presented in Figure 3.4 (a).

We first perform a nosing operation whereby an image is selected from the training dataset, and a time-step $t$ is sampled randomly from a uniform distribution $X \sim U(1, T)$, where $T$ denotes the maximum amount of noising steps. We set $T = 1000$, as is standard practice [12][13][14]. $t$ steps of noise are then added to the image following the noising schedule defined in Section 2.3.3. The noised image is subsequently passed to the diffusion network, alongside a sinusoidal embedding for $t$, and a paired audio embedding produced by the encoding network outlined in Section 3.2. The models objective is to predict the noise added between time-step $t-1$ and $t$.

We evaluate performance by calculating the Mean Squared Error (MSE) between the predicted noise and actual noise added to the image. Finally a gradient descent step is taken using the Adam optimizer, updating model parameters iteratively to minimise the MSE loss.

**Algorithm 1** Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

(a) Training algorithm [7]

**Algorithm 2** Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

(b) Sampling algorithm [7]

Figure 3.4: Training and sampling algorithms used by our diffusion model. Images taken from [7].

After training, the network can be used to generate new audio conditioned images. Again, we follow the sampling procedure outlined by Ho et al. [26]. Pseudo-code for this algorithm can also be found in Figure 3.4 (b).

Firstly, random noise is sampled from an isotropic Gaussian distribution. This sample shares the same dimensionality as the desired output ($64 \times 64 \times 3$) and is the starting point for the reverse noising process. The sample is then processed iteratively by the network $T-1$ times, at each point predicting what the image would look like at the prior time-step. We incorporate conditioning information in the same manner described in training. By following this sampling procedure, the trained diffusion model iteratively removes noise

from the sample, generating a structured output that could plausibly have arisen from the training distribution [26].

We also incorporate classifier-free guidance (CFG) to complement our sampling procedure. This is achieved by generating unconditional samples 20% of the time during training. 20% is within the range recommended by Ho et al. [28]. Conditioning information is removed, in line with Ramesh et al's implementation [14], by setting all values within an audio embedding to 0. At inference, we generate both an unconditional and conditional image, and scale the interpolation between the two to produce a final output. We use a scaling parameter of 3 for our baseline model, informed by findings from Ramesh et al. [14] and Rombach et al. [12]. This approach has been shown to improve image quality and conditional alignment [28]. We refer readers back to Section 2.3.3 for more detail.

## 3.4 Summary

In this chapter, we have offered an in-depth explanation of the architecture employed in our Audio-to-Image generation framework. We highlighted how we use AudioCLIP's audio-head as our embedding network, and detailed how its training procedure encourages embeddings to exist in CLIP's semantically rich latent space. Furthermore, we delved into the structure of the diffusion network, drawing attention to its architecture, alongside the training and sampling procedures.

We accompanied this description of our framework with justifications for the design choices made. Specifically, we noted how the incorporation of AudioCLIP could bring benefits realized by CLIP in the Text-to-Image domain over to the field of Audio-to-Image generation. Moreover, we argued that the constant optimization criteria inherent to diffusion models can make them better behaved, more scalable, and less prone to mode collapse than GANs.

This is the first implementation of both AudioCLIP and a diffusion network within the Audio-to-Image generation literature. We believe that the reasons provided above are sufficient to justify this novel approach.

# Chapter 4

# Experimental Design

## 4.1 Chapter Overview

This chapter details the experimental analyses conducted in this research project. We outline experimental conditions explicitly, alongside relevant hypotheses. This is done to provide readers with a comprehensive understanding of the scientific methodologies employed. Moreover, any interested parties can utilize use this information to conduct their own replication studies.

To this end, we first elaborate upon the dataset used. This includes an outline of its contents and all pre-processing steps applied. Following this, we discuss the individual experiments conducted. For each, we present information regarding relevant hyper-parameters, training durations and evaluation procedures. By doing so, we ensure transparency within our approach, and create a solid foundation for future research.

## 4.2 Dataset

We opted to use the SubURMP dataset created by Chen et al. to facilitate this study [2]. The dataset is comprised of a curated subset of the larger URMP (University of Rochester Multi-Modal Music Performance) dataset [49].

The training set provided contains 71,230 audio-image sample pairs, distributed across musical performances from 13 different instruments. Instruments can be broadly categorized into three groups: strings (violin, viola, cello and double bass), woodwind (flute, oboe, clarinet, bassoon and saxophone) and brass (trumpet, horn, trombone and tuba) [49]. The number of samples associated with each instrument is imbalanced, a point we will refer back to later. A table detailing the exact distribution of observations per class is provided in Table 4.1.

The corresponding test set contains 9,575 audio-image sample pairs associated with the same instruments and artists within the training set. Crucially, training and testing sample pairs are derived from *distinct* musical performances. This is done to minimize data leakage between training and testing sets [2].

Audio files are 0.5 seconds long, sampled at a rate of 44 KHz, and stored in WAVE format. A sliding window of 0.1 seconds is used when converting the original recording into audio samples. The window defines the amount of information shared between a sample at time $t$ and $t-1$. This parameterization translates to 0.4 seconds of shared audio between adjacent samples. Images have a native resolution of 1080P $(1920 \times 1080)$, and are taken from the first frame of each video section associated with the paired audio sample.

### 4.2.1 Preprocessing

We create five distinct audio sample lengths from the original dataset. Those being 1 second, 2 seconds, 3 seconds, 4 seconds, and 5 seconds. This was done to facilitate the experiment outlined in Section 4.3.2. We maintain an overlap length of 0.1 seconds from the original dataset to retain a sufficient number of samples at each length.

Images are down-sampled to a resolution of $64 \times 64 \times 3$ to facilitate efficient training. Doing so is standard practice in state-of-the-art diffusion models where super-resolution networks are employed to increase image resolution [12][13][14][15]. We then crop down-sampled images to ensure instruments and performers occupy the majority of each frame. This is achieved manually by identifying image regions, on a per instrument basis, within which relevant information is contained. Doing so works well as performers tend to stay close to their original location throughout performances. By cropping images we reduce the presence of irrelevant information that could potentially weaken the network's performance. An illustration of the preprocessed images can be found in Figure 4.1.

Following this, we removed instances at each audio length condition that were absent of audio signals (i.e. silent). We found good results were achieved when samples with signal amplitudes below 450 were excluded. We then manually removed samples *largely* devoid of audio from the beginning and end of each performance. Information regarding the final quantity of samples per instrument across audio length conditions is presented in Table 4.1.

In addition to the provided training and testing sets, we also derived a validation set to measure performance on unseen instances during training. This was achieved by partitioning samples from the test set into two halves. We allocated the first half to the validation set and the second to the test set. Any instances containing overlapping audio signals were removed to reduce data leakage. Ideally, we would have created testing and validation sets from distinct musical performances. This was unachievable however, as the original validation set contained only a single performance per instrument. We could have taken unique performances from the training set, although doing so would have removed a substantial portion of our training data. Consequently, we decided to settle for this compromise.

We used the validation set for two purposes. Firstly, its audio samples were used to condition image generation periodically during training. Doing so allowed us to monitor the networks progress on unseen instances visually. We also used it to evaluate model performance against the loss function (MSE). This provided us with a numerical measure of performance on unseen instances during training. No feedback concerning performance on the validation set was provided to the network during training. In contrast, the test set was strictly used to condition the generation of new images from the network after the training procedure.

It is worth noting that the validation and testing sets are relatively small in comparison to the training set (refer to Table 4.1). Whilst not ideal, this was deemed non-problematic for two reasons. Firstly, limited intra-class variance exists within the dataset, as images depict the same artists playing the same instruments in slightly different positions. Secondly, the small scale of the test set facilitated a reasonable sampling duration (approximately 24 hours), given the compute required to facilitate this task.

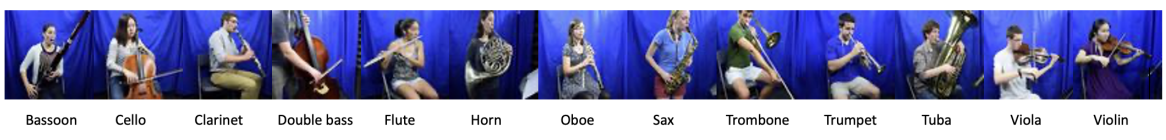



Figure 4.1: $64 \times 64 \times 3$ images of training examples from the SubURMP dataset [2].

Table 4.1: Table of class instances across variants of the SubURMP dataset.

| | Original | | 1 Second | | | 2 Seconds | | | 3 Seconds | | | 4 Seconds | | | 5 Seconds | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Instrument** | **Train** | **Test** | **Train** | **Val** | **Test** | **Train** | **Val** | **Test** | **Train** | **Val** | **Test** | **Train** | **Val** | **Test** | **Train** | **Val** | **Test** |
| **Bassoon** | 1735 | 390 | 1322 | 174 | 163 | 1348 | 171 | 151 | 1358 | 169 | 148 | 1357 | 163 | 142 | 1354 | 158 | 137 |
| **Cello** | 9800 | 1030 | 8549 | 466 | 456 | 8529 | 460 | 440 | 8471 | 460 | 439 | 8529 | 454 | 434 | 8433 | 448 | 427 |
| **Clarinet** | 8125 | 945 | 6972 | 455 | 435 | 6970 | 445 | 425 | 6950 | 446 | 426 | 6947 | 442 | 421 | 6900 | 435 | 414 |
| **Double bass** | 1270 | 1180 | 1223 | 562 | 552 | 1214 | 559 | 538 | 1206 | 555 | 535 | 1196 | 549 | 529 | 1185 | 543 | 523 |
| **Flute** | 5690 | 925 | 5007 | 445 | 435 | 5035 | 438 | 418 | 5033 | 436 | 415 | 5020 | 429 | 409 | 4993 | 424 | 404 |
| **Horn** | 5540 | 525 | 4613 | 239 | 228 | 4849 | 232 | 212 | 4864 | 231 | 210 | 4843 | 224 | 203 | 4798 | 219 | 198 |
| **Oboe** | 4505 | 390 | 3730 | 173 | 163 | 3748 | 172 | 151 | 3735 | 168 | 148 | 3745 | 163 | 142 | 3717 | 158 | 137 |
| **Sax** | 7615 | 910 | 6812 | 154 | 143 | 6832 | 152 | 132 | 6775 | 152 | 132 | 6747 | 148 | 127 | 6643 | 142 | 121 |
| **Trombone** | 8690 | 805 | 7669 | 383 | 372 | 7879 | 383 | 363 | 7893 | 378 | 358 | 7854 | 372 | 351 | 7789 | 366 | 346 |
| **Trumpet** | 1015 | 520 | 991 | 237 | 226 | 975 | 235 | 215 | 972 | 228 | 207 | 962 | 224 | 204 | 953 | 217 | 197 |
| **Tuba** | 3285 | 525 | 3167 | 239 | 228 | 3135 | 232 | 212 | 3124 | 231 | 210 | 3105 | 224 | 204 | 3079 | 220 | 199 |
| **Viola** | 6530 | 485 | 5940 | 233 | 222 | 5800 | 226 | 205 | 5878 | 222 | 201 | 5849 | 217 | 197 | 5795 | 214 | 193 |
| **Violin** | 7430 | 945 | 6876 | 447 | 436 | 6905 | 447 | 427 | 6876 | 442 | 421 | 6856 | 438 | 418 | 6791 | 430 | 410 |

## 4.3 Concurrent Experiments

We begin our investigation by running three experiments concurrently. Those being:

1. Embedding Alignment: A study of how different embedding fine-tuning strategies affect model outputs.

2. Audio Sample Lengths: An evaluation of the impact varying the duration of audio samples has on generative performance.

3. Training Duration: An analysis investigating the impact training time has on generative performance.

We opted to run these experiments concurrently to account for any potential interaction effects between variables. Doing so provides us with a more holistic understanding of model performance, improving the robustness of our analysis.

### 4.3.1 Embedding Alignment

Our investigation into embedding alignment strategies assess how different approaches to fine-tuning the encoding network affect model performance. This is a critical component of the Audio-to-Image generation problem, as the embeddings produced by AudioCLIP define the conditional signal provided to the image generator.

Given Guzhov et al. [6] trained AudioCLIP on relatively small datasets that did not contain instances from SubURMP, we hypothesize that fine-tuning will be necessary to align the network with the current task. To validate this assumption we measure the network's classification performance, before and after applying three different contrastive fine-tuning strategies, those being:

1. Aligning audio embeddings with text embeddings.

2. Aligning audio embeddings with embeddings produced by a frozen image-head.

3. Aligning audio embeddings with embeddings produced by a trainable image-head.

Text-aligned fine-tuning is achieved by first generating an audio and text embedding from the relevant AudioCLIP sub-networks. The audio-head receives an audio sample of length $l$ as input, while the text-head produces its embedding from the associated instrument name. We freeze the text-head during training and contrastively align the audio-head

with it. More specifically, the audio-head is optimized to align its embeddings with text embeddings for the instruments being played. Moreover, the audio-head is simultaneously trained to distinguish audio embeddings from text embeddings produced by other instruments.

This formulation presents a relatively simple learning task, given the constant nature of text embeddings (i.e. only one text embedding is produced per instrument category). While mapping all audio embeddings for a given instrument to a single text embedding may not be possible, it should be feasible to map them to a similar region of the latent space. Doing so should produce separable groups of audio embeddings defined by the source instrument. If so, it should be relatively easy for the generative model to identify instruments based upon their embeddings, and condition the generative process accordingly. This said, the approach does have an associated drawback. Namely, by aligning audio and text embeddings, we may not encode a sufficient amount of visually meaningful information into our conditions. This could potentially lead to the generation of lower quality images.

Considering this, we also align audio embeddings with those produced by a *frozen* image head. We use the term frozen to reference the fact that the image-head weights remain fixed during fine-tuning. In other words, they remain identical to the weights learnt by CLIP [17]. In this implementation the audio sub-network receives an audio sample of length $l$, while the image sub-network receives a $64 \times 64 \times 3$ image from the training set. Here, audio embeddings are aligned with image embeddings from their training sample pair, and shifted away from image embeddings associated with other instruments. As discussed previously (Section 3.2.2), AudioCLIP's image-head has demonstrated a capacity to produce highly visually meaningful embeddings [17]. As such, we reason that this alignment process will result in more visually relevant information being encoded into audio embeddings. However, we also hypothesize that this may be a more challenging task to learn, as unlike text embeddings, image embeddings vary within each instrument class. Moreover, whilst expressive, the image-head, was not trained on SubURMP, thus its representations may not be optimal.

This directs us to our final alignment strategy. In this case, we align audio embeddings with a *mutually* trained image-head. By back-propagating gradients from both image and audio sub-networks we allow both heads to be fine-tuned simultaneously. We expect this to improve the image-head's ability to represent images within the SubURMP dataset. This said, the mutual training of both heads may also result in the loss of some visually expressive information, as image representations are also shifted to align with audio embeddings.

We train AudioCLIP for five epochs with a batch size of 10 across all fine-tuning strategies and audio sample lengths discussed in Section 4.2.1. We further apply stratified sampling to address the class imbalance present in the dataset. During training we optimize AudioCLIP by minimizing the contrastive loss outlined in Section 3.2.2. After fine-tuning, we use learnt audio embeddings as conditioning information for the diffusion image generator. We train the baseline model outlined in Section 3.3.2 for 100 epochs with a batch size of 12, saving model checkpoints every 10 epochs. *We employ this hyper-parameter configuration throughout all of our experiments*.

Model performance is evaluated under three conditions. Firstly, we compare the classification accuracy of each AudioCLIP variant on the SubURMP dataset. This is achieved by calculating the dot product similarity between a given audio embedding and text/image embeddings for each instrument category. The highest dot-product is taken as the prediction. Doing so accords with how previous work has measured the stand-alone performance of their audio encoding network [2][4]. We evaluate image quality using the Frechet Inception Distance (FID)[31] between generated samples and training/testing examples. We further evaluate the extent generated images adhere to conditioning information by incor-

porating an additional image classifier. The classifier is a CNN that is trained to predict the instrument contained within a given image, and achieved 100% accuracy on its testing set. For more details on this approach, we refer readers back to Section 2.4.3.

### 4.3.2 Sample Lengths

Our next concurrently run experiment evaluates model performance across varying audio sample lengths. Previous work by Wen et al. [50] found that as audio sample lengths increase, generated images generally converge upon a correctly aligned output. Their investigation evaluated performance on a paired speech-face dataset, employed a GAN image generator, and considered audio sample lengths of 1, 2, 3, 5 and 10 seconds. We want to know if the same effect can be replicated on the SubURMP dataset with a diffusion image generator.

Our experimental setup faced limitations due to computing constraints. Specifically, we were unable to evaluate 10 second samples, as the associated WAVE files were found to consume an excessive amount of storage, and necessitate an infeasible amount of processing time. As a result, we opted to explore 1, 2, 3, 4, and 5 second sample lengths. Moreover, we believe excluding 10 second samples from our experiment is a reasonable decision irrespective of operating constraints. This is predicated upon the limited differences observed in generated images conditioned upon 5 and 10 second audio samples present in Wen et al.'s [50] work.

We generate audio embeddings for all sample lengths using the three fine-tuned AudioCLIP variants described in the aforementioned experiment. The diffusion model outlined in Section 3.3 is employed as the image generator. We use FID to measure image quality, and quantify conditional adherence with the instrument classifier outlined in 4.3.1.

### 4.3.3 Training Duration

Our final concurrent experiment evaluates the effect training time has on generated samples. We do so by training the baseline conditional diffusion model for 100 epochs, saving weights every 10. In total, we train 15 diffusion models, one for each sample length and fine-tuning strategy combination. We generate images for each variant by conditioning upon audio samples contained within the test set. We conduct 10 sampling runs for each variant, corresponding to each of the saved weights. This translates to 150 sampling iterations. Finally, we evaluate performance by measuring FID scores and instrument classification accuracy on generated images. This is done in the same manner described above in Section 4.3.2.

## 4.4 Classifier-Free Guidance

While the first three experiments are run concurrently for increased robustness, computing constraints made doing so impractical for future experiments. As such, we opted to evaluate the effect of classifier-free guidance (CFG) using embeddings produced by the four-second text-aligned AudioCLIP variant. These embeddings were found to elicit optimal conditional adherence and strong FID scores (see Sections 5.2.2 and 5.2.3). Given CFG is predominantly used to improve conditional alignment (i.e. constancy between generated images and conditioning information) [28], it made sense to conduct this experiment with the best performing model upon this metric.

As a reminder, CFG, outlined in Section 2.3.3, has shown significant potential for improving the semantic alignment between generated samples and conditioning information

[28]. The process involves training a diffusion model to generate both conditional and unconditional images. This is achieved by removing conditioning information $p$-proportion of the time during training. When sampling, the model produces both a conditional and unconditional image. A linear interpolation is taken between the unconditional and conditional images and scaled by a parameter $s \geq 1$. This pushes the unconditional sample toward and beyond the conditional representation, and has been shown to produce a more conditionally faithful image [14][28]. There is however no consensus within the literature for what constitutes an optimal value for $s$. Most popular Text-to-Image diffusion models employ relatively low values (e.g. $s \leq 5$) [12][14], however, some works have found higher values to produce better results [13]. As such, we chose to investigate what effect varying the scaling parameter $s$ would have on the current problem.

To evaluate this, we trained two diffusion models with embeddings from the four second text-aligned AudioCLIP model. We change only the probability of learning to generate unconditional samples $p$ in each model. In the control we set $p = 0$ such that the model never generates unconditional samples. In the CFG enabled case, we set $p = 0.2$, which is within the range recommended in the original paper [28]. We then sample from both models, conditioning upon test set audio embeddings. We do so six times in the CFG enabled case, varying $s$ each time. We specifically evaluate CFG at $s = 1, 3, 5, 7, 9, 11$. These values were chosen to explore a range of both low and high scaling parameters. We once again use FID scores and instrument classifier accuracy to evaluate model performance.

## 4.5 Projection Mechanism

In our final experiment we investigate two different techniques for reducing the dimensionality of audio embeddings. We specifically compare the performance of a learnable leaky ReLU activated fully connected layer to the results achieved via Principal Component Analysis (PCA).

Our encoding model produces high dimensional (1024 dim) embeddings [6]. It is common practice within the literature to reduce these, as doing so can lower the compute required for training, and allow the generative model to distil pertinent information [14][12][2]. Fully connected approaches are common within the Audio-to-Image generation domain [2][51], although PCA reduction has yet to be evaluated. We consider doing so reasonable given the successful application of PCA reduction in DALL-E 2 [14].

We evaluate the two conditions using four second text-aligned audio samples, and train the models for 100 epochs, varying only their projection mechanisms. Once again, FID and instrument classification accuracies are used to evaluate the quality of generated samples conditioned upon test set audio embeddings.

# Chapter 5

# Results and Discussion

## 5.1 Chapter Overview

In this chapter, we present the findings associated with the experiments detailed in Chapter 4. To begin, we analyse the results of those experiments run concurrently. This is followed by the results associated with our studies into classifier-free guidance (CFG) and projection mechanisms. We empirically analyse our results and discuss their relevance within the broader context of the domain. By doing so, we provide insights into why they were observed and their implications for the field of Audio-to-Image generation.

## 5.2 Concurrent Experiments

In this section, we describe the results of our concurrently run experiments. We structure this by order of evaluation criteria, and discuss all three experiments collectively. Presenting information in this manner makes the most sense as these results must be considered holistically.

The three experiments investigated the conjoint effect different fine-tuning strategies, audio sample lengths, and training durations have on model performance. Three fine-tuning strategies were considered, whereby AudioCLIP's audio-head was aligned with either its text-head, frozen image-head, or mutually trained image-head. We refer to these strategies as *text*, *frozen* and *mutual* respectively. Our analysis of audio sample lengths considered the effect different audio sample durations have on model performance. Those durations being 1, 2, 3, 4 and 5 seconds. Finally, to assess how training duration impacts performance, we generated images conditioned upon withheld audio samples for each model using weights saved every 10 epochs throughout the 100 epoch training duration.

We evaluated model performance on upon three criteria. The first, pertaining only to the fine-tuning study, measured the classification performance of the fine-tuned AudioCLIP models on the SubURMP dataset. The second metric uses a pre-trained instrument classifier to measure how well generated samples align with conditioning information. Finally, we incorporated FID scores as a quantitative measure of image quality.

### 5.2.1 AudioCLIP Classification Performance

Given each AudioCLIP sub-network produces an embedding rather than a classification prediction, we are forced to diverge from the conventional evaluation of classification algorithms. We opt to employ AudioCLIP's querying functionality as a proxy. This was described briefly in Section 4.3.1, although in the interest of clarity, we provide more detail

here.

Querying is achieved by first deriving an audio embedding for a given sample. A query-set is then produced, containing a list of embeddings from the alignment modality (either text or image). Each embedding within the query-set is associated with a different instrument class. In the text-aligned case, this translates to a set of embeddings where each is associated with a different instrument name. In the image-aligned case, the query-set consists of the image embedding for the truly associated image, and embeddings from a randomly selected instance from each of the other instrument classes. The dot product between the audio and all query embeddings is then calculated, measuring their similarity. The query embedding associated with the largest dot product is taken as the model prediction.

We provide the raw querying accuracies in Table 5.1, and an associated visualization in Figure 5.1.

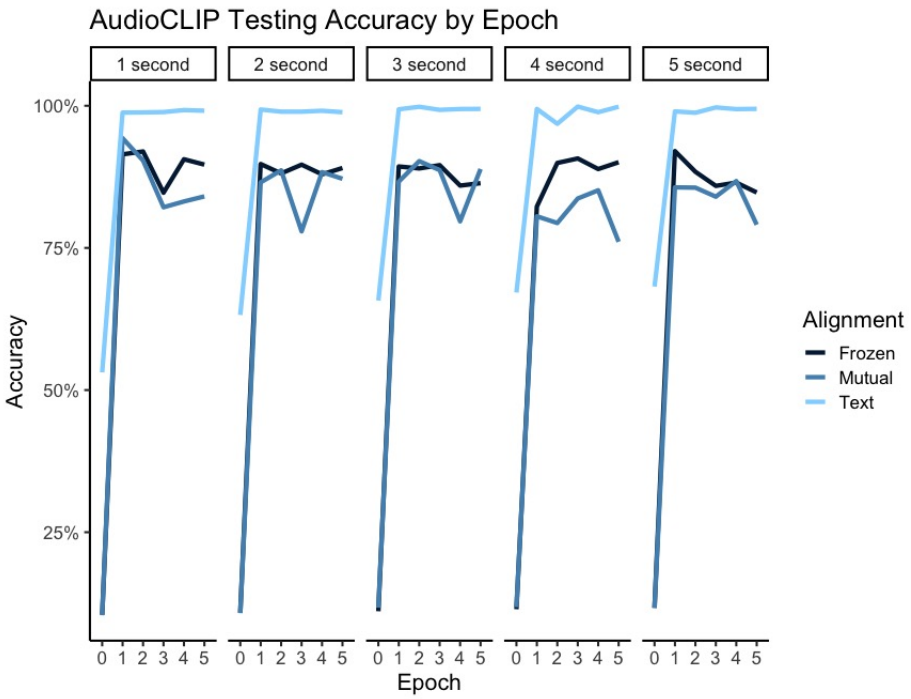



Figure 5.1: AudioCLIP testing accuracy by epoch.

**Zero-shot performance**

The largest performance discrepancy present in the results, occurs prior to any fine-tuning at epoch 0. This quantifies AudioCLIP's zero-shot performance on SubURMP, and was calculated to validate the necessity of applying fine-tuning (see Section 4.3.1, Paragraph 2). By zero-shot, we refer to the model's capacity to generalize to instances outside of its training dataset [14]. We note that both image-aligned models can be considered identical at epoch 0, given no training has occurred.

Under these conditions, we see that text-alignment elicits substantially better performance than image-alignment. Specifically, achieving a classification accuracy of 53% with 1 second audio samples, compared to the image variants 10%. We note that the image variants performance is only 2.3% better than random guessing under stratified sampling conditions.

A monotonic increase in classification accuracy is observed for all models as audio sample lengths increase. This improves text-aligned performance to 68% and image-aligned per-

formance to 12% with five second samples. We found these improvements to be associated with strong, positive and significant correlations at $\alpha = 0.05$ for both querying variations. Specifically, text-aligned querying returned the following statistics $(R = 0.88, p = 0.048, n = 5)$, while image-aligned querying returned $(R = 0.85, p = 0.002, n = 10)$. The lower $p$-value in the image-aligned case is a result of the larger sample size. We emphasize that the comparable $R$ values between tests are reflective of the similar proportional increases observed, *not* the magnitude of those increases.

While both image models can be considered identical at epoch 0, there exists a small disparity in their performance when three and four second samples are used (see Table 5.1). We attribute this to the variance introduced via the random selection of images necessary to facilitate dot product calculations. Whilst not ideal, this should not have a significant impact on our analysis given the resounding similarity between instances of the same class.

Taken collectively, results indicate that zero-shot text-aligned querying is substantially more effective than image querying. We reason this discrepancy is a result of two factors. Firstly, the consistent intra-class text embeddings remove a source of variation inherent to image querying. Secondly, images within the SubURMP dataset bear significant resemblance to one another, irrespective of the instrument being played. For instance they all contain blue backgrounds and have an individual occupying the center of the frame (see Figure 4.1). This similarity may make it challenging for the image-head to separate instruments within the latent space zero-shot. If so, querying would necessarily become more challenging.

In addition, it was found that increasing audio-sample lengths universally improves zero-shot performance. We posit that the addition information longer sample lengths contain, provide are a plausible explanation for this trend. Finally, we note that the classification performances achieved highlight the need for fine-tuning. Previous works have achieved near perfect accuracies (e.g. 99.99% [4]) on this metric. It is therefore, critical that we find a means of improving performance, to place our work on par with these benchmarks.

Table 5.1: AudioCLIP testing classification accuracy.

| Time | Alignment | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|
| 1 second | Frozen | 0.10 | 0.91 | **0.92** | 0.85 | 0.91 | 0.90 |
| 2 second | Frozen | 0.11 | **0.90** | 0.88 | 0.90 | 0.88 | 0.89 |
| 3 second | Frozen | 0.11 | 0.89 | 0.89 | **0.90** | 0.86 | 0.86 |
| 4 second | Frozen | 0.11 | 0.82 | 0.90 | **0.91** | 0.89 | 0.90 |
| 5 second | Frozen | 0.12 | **0.92** | 0.88 | 0.86 | 0.86 | 0.85 |
| 1 second | Mutual | 0.10 | **0.94** | 0.90 | 0.82 | 0.83 | 0.84 |
| 2 second | Mutual | 0.11 | 0.87 | **0.89** | 0.78 | 0.88 | 0.87 |
| 3 second | Mutual | 0.12 | 0.87 | **0.90** | 0.89 | 0.80 | 0.89 |
| 4 second | Mutual | 0.12 | 0.81 | 0.79 | 0.84 | **0.85** | 0.76 |
| 5 second | Mutual | 0.12 | 0.86 | 0.86 | 0.84 | **0.87** | 0.79 |
| 1 second | Text | 0.53 | **0.99** | 0.99 | 0.99 | 0.99 | 0.99 |
| 2 second | Text | 0.63 | **0.99** | 0.99 | 0.99 | 0.99 | 0.99 |
| 3 second | Text | 0.66 | 0.99 | **1.00** | 0.99 | 0.99 | 0.99 |
| 4 second | Text | 0.67 | 0.99 | 0.97 | **1.00** | 0.99 | 1.00 |
| 5 second | Text | 0.68 | 0.99 | 0.99 | **1.00** | 0.99 | 0.99 |

**Fine-tuned performance**

By fine-tuning AudioCLIP, we were able to achieve substantially better results. Specifically, the text-aligned model achieved near-perfect classification accuracies (99%) on withheld data after only one training epoch. This was observed over all sample lengths. Performance further improved to 100% when trained for longer durations on increased samples lengths (3, 4, and 5 seconds). This said, the associated correlation between audio sample lengths and text-aligned accuracy narrowly precluded significance at $\alpha = 0.05$, returning the following results: $(R = 0.87, p = 0.058, n = 5)$.

Both image-aligned models also realised substantial improvements to their zero-shot performance post fine-tuning. Specifically, the best performing frozen model classified 92% of test instances correctly when two and five second audio samples were used. The mutually aligned network improved this slightly, classifying 94% of instances correctly with one second audio samples. No significant correlation between sample length and classification performance was found for either model at $\alpha = 0.05$.

We found a highly significant relationship between optimal training time and audio sample lengths across all models $(R = 0.63, p = 0.01, n = 15)$. Thus suggests that a moderately strong, positive linear relationship exists between the two variables. In other words, longer audio sample lengths generally require longer training durations to achieve optimal results.

We note that the classification performances of image-aligned models are more varied across training epochs than the text-aligned model. This could indicate that image-aligned fine-tuning is less robust, but could equally be attributed to the necessarily stochastic nature of image evaluation. We would have liked to run more experiments to answer this statistically; however, computing limitations made this infeasible.

These results support the conclusion that fine-tuning AudioCLIP is necessary to achieve good performance on the SubURMP dataset. Moreover, they show that text-aligned performance continues to surpass image-aligned performance even after fine-tuning. As before, we reason that the static nature of intra-class text embeddings makes it an inherently easier task than image-alignment. This said, fine-tuned image-aligned models performed well, with little discrepancy between *frozen* and *mutual* training strategies. They were, however observed to exhibit greater variation in performance across training durations. This could suggest that image-alignment is inherently less robust, however, our result do not allow us to make any firm conclusions.

We note that no significant relationships were found between audio sample lengths and classification performance on fine-tuned models. This may suggest that the additional information provided by longer sample lengths is less useful after fine-tuning. That said, we are working with a particularly small sample size which could make detecting these patterns difficult. It would be worth while for future research to conduct a similar analysis with additional samples to help address this question.

### 5.2.2 Generated Sample Classification Performance

In this section, we shift focus to evaluating the extent to which generated images align with conditioning information. We refer to this as *conditional adherence*. To do so, we train fifteen versions of the diffusion model described in Section 3.3, each conditioned upon audio embeddings from a unique fine-tuning, sample length combination. We reiterate here that embeddings are produced by AudioCLIP's audio-head. Text-and-image-heads were only used for the purposes of fine-tuning.

After training we generate images from each diffusion model, conditioning upon test set audio samples. Each sample is converted into an embedding via the same fine-tuned audio-head used during the diffusion models training procedure. We measure conditional

adherence by incorporating a pre-trained CNN capable of recognising instruments within images with 100% accuracy on unseen instances (See Sections 2.4.3 and 4.3.1 for more details).

**Model level performance**

Analyzing the results displayed in Figure 5.2 reveals several interesting trends. Firstly, images generated by models conditioned upon text-aligned audio embeddings tend to be more aligned with conditioning information than those conditioned upon image-aligned embeddings. This is evidenced by the higher instrument classification accuracies shown in text-aligned cases (see Table 5.2 for exact values). The only exception to this rule being when five second audio samples are used. Moreover, results show that the best performing model was trained using four second text-aligned embeddings for 70 epochs, yielding a classification accuracy of 68.66%. Notably this is lower than state-of-the-art GANs, the best of which achieves 89% on the same metric.

In addition, results indicate that models conditioned upon frozen image embeddings tend to outperform those conditioned upon mutually aligned embeddings. The only exception to this being observed with three second audio samples.

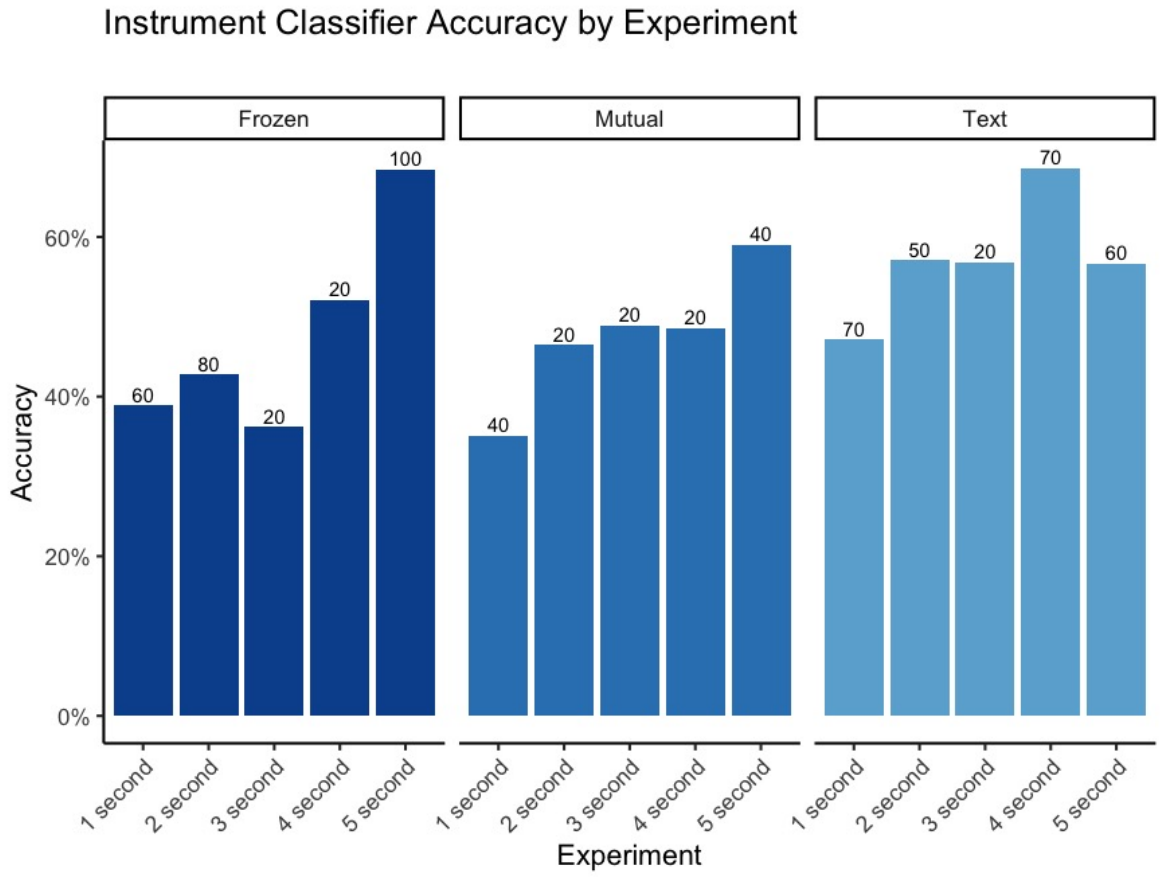



Figure 5.2: Instrument classification accuracy by experiment. Numbers above bars report the best performing epoch.

With regard to sample lengths, all variations (text, frozen, mutual) show a generally improving trend in conditional adherence as audio sample lengths increase. This is most pronounced in image-aligned models, with both variants achieving their best performance

when conditioned upon five-second samples (frozen = 68.4%, mutual = 58.01%)

Optimal training durations vary across different models, with some reaching peak performance after only 20 epochs, and others requiring 100 (see Table 5.2). This said, performance typically plateaus after 20 or 30 epochs, after which classifier accuracies oscillate, changing by only a few percentage points in either direction.

Table 5.2: Instrument classifier results by epoch.

| Time | Alignment | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Frozen | 17.76% | 38.68% | 38.19% | 38.73% | 37.74% | **38.88%** | 38.83% | 38.11% | 38.29% | 38.38% |
| 2 | Frozen | 19.72% | 30.29% | 42.66% | 41.17% | 42.71% | 42.58% | 42.61% | **42.74%** | 41.22% | 41.96% |
| 3 | Frozen | 28.91% | **36.18%** | 23.53% | 23.35% | 23.64% | 24.08% | 24.16% | 23.95% | 23.14% | 23.27% |
| 4 | Frozen | 29.01% | **52.05%** | 37.58% | 37.13% | 38.69% | 37.87% | 37.34% | 38.43% | 37.74% | 37.64% |
| 5 | Frozen | 37.21% | 24.50% | 68.08% | 67.35% | 66.92% | 67.32% | 67.22% | 66.32% | 67.49% | **68.40%** |
| 1 | Mutual | 13.99% | 34.81% | 34.39% | **35.11%** | 34.47% | 33.53% | 33.41% | 34.29% | 35.01% | 34.57% |
| 2 | Mutual | 22.40% | **46.44%** | 40.50% | 39.52% | 39.39% | 39.26% | 38.65% | 39.93% | 39.75% | 40.47% |
| 3 | Mutual | 21.74% | **48.94%** | 43.82% | 45.12% | 44.13% | 44.70% | 44.49% | 43.90% | 44.94% | 43.53% |
| 4 | Mutual | 31.16% | **48.59%** | 45.97% | 45.20% | 45.68% | 45.57% | 45.36% | 44.88% | 45.54% | 45.65% |
| 5 | Mutual | 22.91% | 47.09% | 57.83% | **59.01%** | 58.34% | 58.34% | 57.77% | 58.45% | 58.96% | 57.85% |
| 1 | Text | 16.97% | 34.98% | 46.05% | 47.08% | 46.51% | 47.06% | **47.15%** | 47.06% | 46.86% | 46.32% |
| 2 | Text | 26.97% | 32.22% | 55.18% | 54.46% | **57.08%** | 57.01% | 56.49% | 56.70% | 54.62% | 55.41% |
| 3 | Text | 34.08% | **56.81%** | 54.75% | 55.66% | 55.12% | 55.14% | 55.58% | 55.35% | 55.48% | 54.34% |
| 4 | Text | 43.03% | 63.18% | 67.02% | 67.63% | 67.68% | 68.13% | **68.66%** | 67.23% | 67.79% | 66.89% |
| 5 | Text | 21.26% | 33.32% | 56.42% | 55.77% | 56.10% | **56.66%** | 56.53% | 56.23% | 55.77% | 56.48% |

To contextualize these findings, we argue the following. Firstly, generative models conditioned upon text-aligned audio embeddings tend to generate more conditionally faithful samples. This is likely due to the consistent intra-class text embeddings used in the text-aligned fine-tuning process, making it easier to separate audio conditions into distinct regions of the latent space. By doing so, it becomes easier for the generative network to interpret the contents of the embedding.

Findings also indicate that diffusion models trained on frozen image-aligned embedding, tend to align with conditioning information better than those conditioned upon mutually-aligned embeddings. A plausible explanation for this is that by freezing image-head weights during fine-tuning, we maintain the full expressiveness of the image-head. In contrast when image-head gradients are enabled, the image-head shifts to align with audio embeddings. By doing so, some of the visual expressiveness of the image-head may be lost in the process.

We noted that performance generally improves for all models when audio sample lengths are increased. This trend being most clearly exhibited by diffusion models conditioned upon image-aligned audio embeddings. We attribute this trend to the additional information provided by longer audio samples. The disproportionate benefit realised by image-aligned variants may be due to the increased complexity of the task, whereby the additional information may prove more beneficial. It would be interesting to investigate whether even longer audio samples could provide further benefit as performance did not appear to plateau in either image-aligned case.

Interestingly, the advantages provided by longer audio samples were not reflected in the previous experiment when measuring fine-tuned AudioCLIP classification accuracies (see Section 5.2.1). Moreover, AudioCLIP classification accuracies were considerably more favourable than the observed conditional adherence scores. This implies that the classification performance of the embedding network, is not necessarily indicative of performance on the downstream generative task. This is particularly noteworthy as previous research has often used to measure to evaluate the quality of their encoding netowkrs [2][4].

**Instrument level performance**

To gain a more complete understanding of the diffusion model's adherence to conditioning information, we computed a class level confusion matrix with our best performing model. That being, the diffusion generator trained for 70 epochs conditioned upon four second text-aligned embeddings. Readers can find the confusion matrix in Figure 5.3.

The matrix is best read from left to right. Values in each row represent predictions made by the instrument classifier for each true image label. For instance, the bottom right-most value provides the proportion of violin images that were generated from bassoon audio samples. Values along the diagonal show the proportion of correctly aligned samples. A well performing model would have high values along the diagonal and low values elsewhere.

We identified three performance categories based upon the proportion of correctly classified instances for each instrument. Those being Poor ($\leq 0.3$), Moderate ($0.3 < 0.9$), and High ($\geq 0.9$). The list of instruments attributed to each category is presented below:

- **High**: Viola, Tuba, Trombone, Double bass.
- **Moderate**: Bassoon, Cello, Clarinet, Flute, Viola, Trumpet.
- **Poor**: Horn, Oboe, Saxophones.

An interesting trend is shown on the confusion matrix for those instruments in the poorly aligned category. Specifically, we notice that oboes and saxophones are frequently misclassified as flutes. This indicates that oboe and saxophone conditions frequently result in generated images of flutes. Importantly, all three of these instruments are woodwinds [49]. Furthermore, in Figure 5.4, we can see that their associated images appear visually similar to one another. They are all long and thin, held is comparable ways, and being blown into by the performer. Similarly, horns were routinely misclassified as either trumpets or saxophones, both of which are made from brass. Taken collectively, this indicates that whilst the model makes mistakes, it does so in an understandable manner, confusing instruments that share significant similarities to one another.

### 5.2.3 FID Scores

The final aspect of our concurrent analysis regards FID scores. These allow us to quantify the visual quality of generated samples (see Section 2.4.2). We evaluate generated images against true training and testing images and present results in Tables 5.4 and 5.5 respectively. We note that images within the training and testing datasets are relatively similar to one another. This is an inherent limitation of the dataset used and is discussed in more detail in Chapter 6.

Results indicate that the best testing FID scores were achieved by the diffusion model conditioned on three second, mutually-aligned audio embeddings. Specifically, eliciting a FID score of 75.32. This is markedly better than existing GAN solutions on the same dataset. The best of these alternatives, created by Chung et al. [2], achieved 107.26. See Table 5.3 for more detail. We note that Chung et al.'s model generated larger resolution images than ours ($256 \times 256 \times 3$), limiting the conclusions that can be drawn from this. That said, our model improves FID scores by over 70% compared to S2I-C [2], the only other work generating $64 \times 64 \times 3$ images.

Generated images conditioned upon four and five second text-aligned embeddings also returned impressive FID scores, precisely achieving 76.11 and 75.11 respectively. This suggests that comparable levels of image quality can be realised with both conditioning ap-

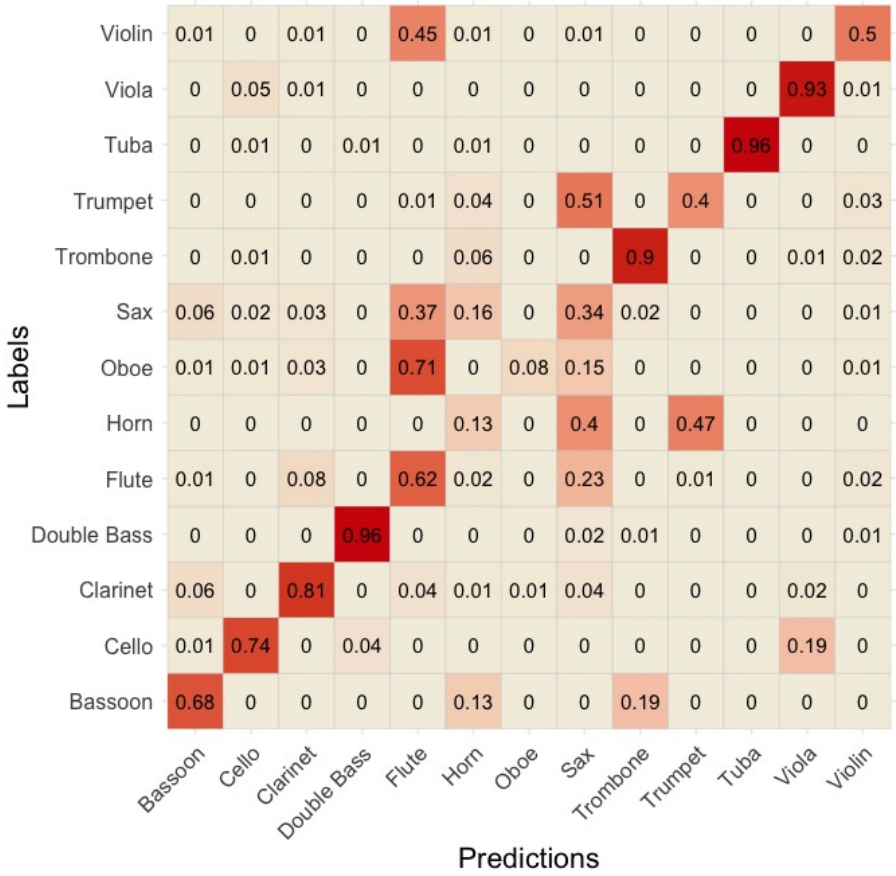



Figure 5.3: Confusion matrix of instrument classifier results on 4 second text conditioned generated images. Values are proportions.

proaches. The same cannot be said for frozen image embeddings. The best of which, conditioned on 1 second audio samples, achieved an FID score of 84.94.

Networks conditioned upon text-aligned and frozen image-aligned embeddings display a generally improving trend in FID scores as audio sample lengths increase. The same trend is not observed for diffusion networks conditioned upon mutually-aligned embeddings. More precisely, mutually conditioned models show a worsening in FID performance when sample lengths exceeded three seconds. These trends can be best observed in Figure 5.5. Given the limited number of samples, it is difficult to determine how representative this trend is. Nonetheless, it may imply that mutually-aligned embeddings are associated with less predictable performance than embeddings fine-tuned against a fixed alignment head.

With regard to training time, better FID scores are generally associated with lower epoch counts for both image-aligned variants. In contrast, models conditioned upon text-aligned embeddings appear to require slightly more training durations to achieve maximum performance.

In line with expectations, training FID scores were found to be consistently better then testing FID scores. This suggests that the generated images produced bear greater resemblance to the training distribution. Nonetheless, the difference between scores are relatively small, with little variance observed. Specifically, the mean difference between measures was 15.58 units with an associated standard deviation of 4.64.

In summary, our models outperformed GANs with regard to FID. We preface this finding by noting that, state-of-the-art GANs generate higher resolution $256 \times 256 \times 3$ images

(a) Oboe (b) Saxophone (c) Flute

Figure 5.4: Poorly semantically aligned image examples from the pre-processed SubURMP dataset.

Table 5.3: FID comparison between state-of-the-art GAN Audio-to-Image generation networks [4].

| Method | FID |
|---|---|
| S2I-C | 252.66 |
| CMCGAN | 215.92 |
| CMCRL + SAGAN | 107.26 |
| Best Diffusion | 75.32 |

[4]. As a result, it is not entirely fair to compare the two measures. In Chapter 6 we describe how future work could consider employing super-resolution networks to address this concern. This said, the original S2I-C network [2] did produce $64 \times 64 \times 3$ samples, and our network vastly outperforms this approach (see Table 5.3). We reason that this improvement is due to FID's propensity to reward both high fidelity and diverse image samples [28]. GANs are known to suffer from mode collapse [27], whereby they only learn to generate samples from a limited subset of the output distribution. In short, samples from GANs can lack diversity. In contrast, diffusion models do not suffer from the same problem, learning to produce more diverse samples, potentially helping improve FID scores [27].

Our best performing model utilized 3 second mutually-aligned embeddings, although this was closely followed models conditioned upon 4 and 5 second text-aligned embeddings. Models conditioned upon text-aligned embeddings generally showed greater consistency in FID scores across embedding lengths, indicating they may be a more robust solution to the problem.

Models conditioned upon frozen image embeddings achieved the least satisfactory FID scores. We believe this performance gap could be attributed to the image-head not being sufficiently aligned with the SubURMP dataset. By fine-tuning image embeddings, we reason that the image-head becomes better able to represent details in SubURMP images. This would explain the better results observed when mutually-aligned conditions were used.

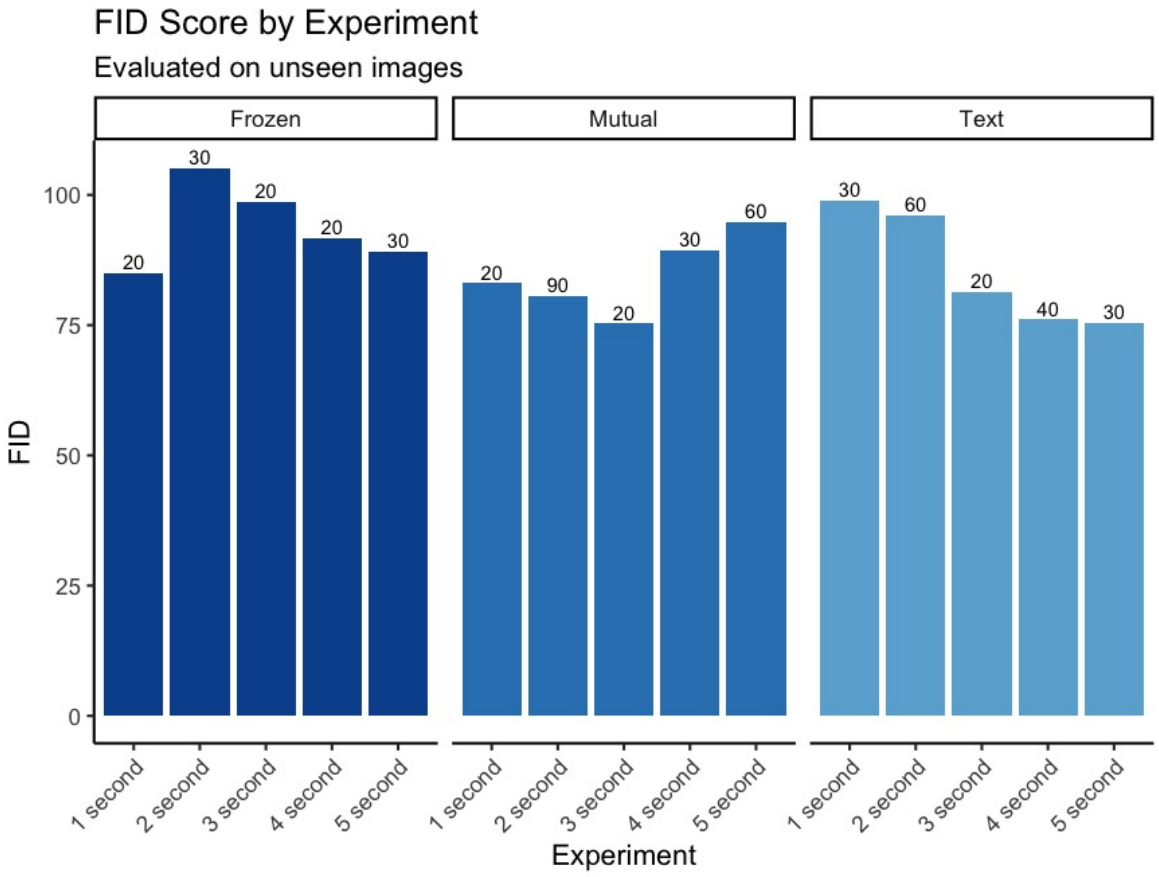



Figure 5.5: FID scores on testing instances. Numbers above bars represent the best performing epoch.

Table 5.5: FID testing scores by epoch.

| Time | Alignment | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Frozen | 138.93 | **84.94** | 95.18 | 96.22 | 96.73 | 96.63 | 96.63 | 96.73 | 96.22 | 95.18 |
| 2 | Frozen | 124.92 | 114.56 | **105.21** | 107.26 | 106.95 | 107.86 | 107.86 | 106.95 | 107.26 | 105.21 |
| 3 | Frozen | 125.97 | **98.69** | 109.42 | 111.11 | 109.58 | 110.92 | 110.92 | 109.58 | 111.11 | 109.42 |
| 4 | Frozen | 123.28 | **91.69** | 99.80 | 99.59 | 99.14 | 98.89 | 98.89 | 99.14 | 99.59 | 99.80 |
| 5 | Frozen | 121.74 | 133.09 | **89.07** | 90.34 | 91.02 | 89.71 | 89.71 | 91.02 | 90.34 | 89.07 |
| 1 | Mutual | 150.34 | **83.04** | 90.86 | 90.45 | 91.12 | 90.87 | 90.87 | 91.12 | 90.45 | 90.86 |
| 2 | Mutual | 119.08 | 86.24 | 84.08 | 80.85 | 81.10 | 82.42 | 81.95 | 82.47 | **80.63** | 82.24 |
| 3 | Mutual | 121.56 | **75.32** | 103.08 | 100.71 | 100.92 | 101.62 | 101.62 | 100.92 | 100.71 | 103.08 |
| 4 | Mutual | 119.21 | 99.74 | **89.34** | 91.12 | 89.56 | 91.80 | 91.80 | 89.56 | 91.12 | 89.34 |
| 5 | Mutual | 137.66 | 97.87 | 94.84 | 95.83 | 96.32 | **94.78** | 94.78 | 96.32 | 95.83 | 94.84 |
| 1 | Text | 131.73 | 145.48 | **98.99** | 99.34 | 99.72 | 101.01 | 100.65 | 100.47 | 99.70 | 99.23 |
| 2 | Text | 137.28 | 113.99 | 97.05 | 97.18 | 97.25 | **96.08** | 96.08 | 97.25 | 97.18 | 97.05 |
| 3 | Text | 114.23 | **81.47** | 97.04 | 95.94 | 94.88 | 96.38 | 96.38 | 94.88 | 95.94 | 97.04 |
| 4 | Text | 127.46 | 95.44 | 77.57 | **76.11** | 77.48 | 78.69 | 78.69 | 77.48 | 76.11 | 77.57 |
| 5 | Text | 119.48 | 99.66 | **75.51** | 75.92 | 76.49 | 77.46 | 77.46 | 76.49 | 75.92 | 75.51 |

Table 5.4: FID training scores by epoch.

| Time | Alignment | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Frozen | 125.33 | **73.47** | 80.73 | 81.28 | 82.09 | 81.49 | 81.49 | 82.09 | 81.28 | 80.73 |
| 2 | Frozen | 108.51 | 100.14 | **90.15** | 91.98 | 91.28 | 92.25 | 92.25 | 91.28 | 91.98 | 90.15 |
| 3 | Frozen | 115.86 | **79.44** | 92.56 | 94.96 | 93.17 | 94.34 | 94.34 | 93.17 | 94.96 | 92.56 |
| 4 | Frozen | 106.08 | **78.95** | 84.21 | 83.64 | 83.70 | 83.38 | 83.38 | 83.70 | 83.64 | 84.21 |
| 5 | Frozen | 109.25 | 114.10 | **80.17** | 81.54 | 81.43 | 81.13 | 81.13 | 81.43 | 81.54 | 80.17 |
| 1 | Mutual | 133.57 | **69.27** | 75.51 | 74.62 | 75.38 | 75.16 | 75.16 | 75.38 | 74.62 | 75.51 |
| 2 | Mutual | 104.75 | 70.92 | 64.58 | 61.75 | 61.64 | 63.38 | 62.24 | 63.02 | **61.22** | 62.80 |
| 3 | Mutual | 104.87 | **53.20** | 86.47 | 84.72 | 85.26 | 85.64 | 85.64 | 85.26 | 84.72 | 86.47 |
| 4 | Mutual | 101.84 | 81.37 | **69.25** | 71.17 | 69.09 | 71.84 | 71.84 | 69.09 | 71.17 | 69.25 |
| 5 | Mutual | 123.60 | 79.56 | **75.37** | 76.15 | 76.48 | 75.22 | 75.22 | 76.48 | 76.15 | 75.37 |
| 1 | Text | 115.52 | 130.47 | **81.55** | 81.87 | 82.11 | 83.30 | 82.93 | 83.28 | 82.53 | 81.68 |
| 2 | Text | 118.73 | 100.88 | **82.10** | 82.38 | 82.75 | 81.24 | 81.24 | 82.75 | 82.38 | 82.10 |
| 3 | Text | 96.35 | **62.56** | 80.31 | 79.81 | 78.83 | 79.78 | 79.78 | 78.83 | 79.81 | 80.31 |
| 4 | Text | 111.42 | 79.83 | **66.34** | 65.05 | 66.24 | 68.11 | 68.11 | 66.24 | 65.05 | 66.34 |
| 5 | Text | 100.04 | 80.12 | **64.30** | 65.10 | 64.77 | 65.82 | 65.82 | 64.77 | 65.10 | 64.30 |

## 5.3 Classifier-Free Guidance

In our next study we investigate the effect classifier-free guidance (CFG) and its associated scaling parameter $s$ have on image quality and conditional adherence.

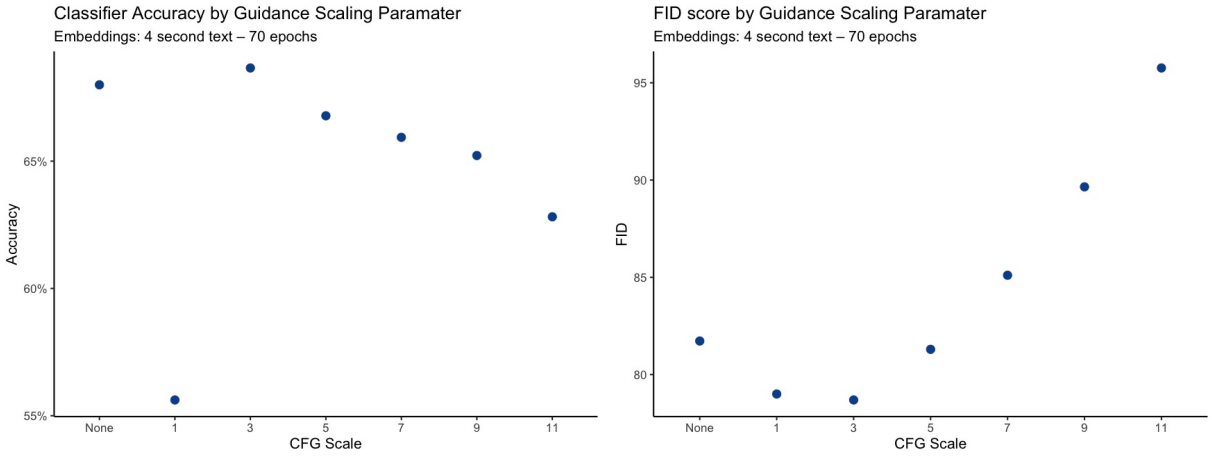
To do so, we train two near identical versions of the diffusion model outlined in Section 3.3. The first is trained to solely generate conditional images, while the second also learns to generate unconditional images 20% of the time. The latter provides the capacity to integrate CFG (see Sections 2.3.3 and 4.4 for more detail). Both networks incorporate embeddings from a text-aligned encoding model utilizing four-second audio samples. We opted to use these embeddings as they elicited the best alignment between generated samples and conditioning information (see Table 5.2) and demonstraited strong FID scores (see Table 5.5). Given CFG is predominantly used to improve conditional adherence [28], it made sense to condition both models on the embeddings that performed best on this metric.

With respect to FID scores, an initial improvement was observed when CFG was enabled compared to when it was not. Further improvements were realised when $s$ was increased to 3. Increasing $s$ beyond 3 was associated with monotonically worsening FID scores. This pattern mirrors those found by Ho et al. [28] and Ramesh et al. [14]. We direct readers to Figure 5.6 (b) for a visualization of this trend, and Table 5.6 for the raw values.

We attribute the observed trend to the inherent trade of between diversity and fidelity present when using CFG [28]. As $s$ is increased, generated images are shifted closer toward the archetypal instance of their class [28]. By doing so, image diversity is necessarily decreased. This said, shifting samples towards an archetypal instance also improves their fidelity, as those samples become more representative of their class [28]. As mentioned previously, FID rewards high diversity and fidelity in generated samples [28][14]. We can see that a balance is struck between these two elements when $s = 3$, explaining the observed pattern.

Table 5.6: FID scores by CFG scaling parameter.

| CFG Scale | None | 1 | 3 | 5 | 7 | 9 | 11 |
|---|---|---|---|---|---|---|---|
| FID | 81.72 | 79.00 | 78.69 | 81.30 | 85.11 | 89.65 | 95.76 |




(a) Accuracy by CFG scale

(b) FID by CFG scale

Figure 5.6: Accuracy and FID scores by CFG scaling parameter.

Interestingly, while increasing $s$ is thought to push samples towards an archetypal instance of their class, we do not observe a constant improvement in instrument classification accuracy when $s$ is increased. Our findings showed relatively high performance without enabling CFG (67.9%), a substantial decrease in conditional adherence when CFG was enabled and $s = 1$ (55.62%), and optimal performance when $s = 3$ (68.66%). As with FID, increasing $s > 3$ resulted in worsening conditional adherence. We refer readers to Figure 5.6 (a) for a visualization of this trend and Table 5.7 for specific values.

We hypothesize that the large drop in performance observed when $s = 1$ is a natural consequence of the reduction in conditional learning when CFG is enabled. When $s = 1$ we are tasking the CFG model to produce its conditional sample without any further guidance. This is because an interpolation between the unconditional and conditional sample, scaled by 1, equates to the conditional sample itself. Thus, our findings here effectively illustrate the loss of performance associated with training the generative model to sample unconditionally 20% of the time.

On the other hand, we believe the decrease in performance observed when $s > 3$ is illustrative of excessively aggressive scaling. In particular, we argue that an overly large interpolation scale could push image representations into areas of the data manifold associated with different classes. As a result, generated images would no longer appear to belong to their original class. To support this hypothesis, we draw reader's attention to Figure 5.7, which displays generated samples that contain components of multiple different instruments. Importantly, these kinds of outputs were only found in cases where large scaling parameters were used. This supports our argument, providing visual evidence to suggest that large scaling parameters shift image representations into areas of the data manifold no longer solely associated with the original condition.

Table 5.7: Accuracy by CFG scale.

| CFG Scale | None | 1 | 3 | 5 | 7 | 9 | 11 |
|---|---|---|---|---|---|---|---|
| Accuracy | 67.9% | 55.62% | 68.66% | 66.78% | 65.93% | 65.22% | 62.81% |

To take a more detailed look into the effect CFG has on conditional adherence, we investigate the amount of correctly classified images on a per instrument basis. Raw figures associated with this analysis are presented in Table 5.8.

(a) Scale = 9 (b) Scale = 11 (c) Scale = 11

Figure 5.7: Generated images containing multiple different instruments.

A noticeable decrease in instances of extremely low instrument level classification accuracies is observed when CFG is disabled. This is best illustrated in Figure 5.8. We further note that as $s$ is increased beyond one, a consistent reduction in the Interquartile Range (IQR) is observed. This suggests that conditional adherence becomes more comparable across instrument classes as $s$ is increased. Importantly, this uniformity appears to come at the cost of overall performance. This is supported statistically by the highly significant negative relationship between $s$ and mean accuracy ($R = -0.969, p = 0.006$).

Table 5.8: Accuracy by CFG scaling parameter.

| Name | None | 1 | 3 | 5 | 7 | 9 | 11 |
|---|---|---|---|---|---|---|---|
| Bassoon | 36.62% | 63.38% | 67.61% | 63.38% | 68.31% | 62.68% | 55.63% |
| Cello | 80.41% | 66.82% | 74.65% | 71.89% | 70.05% | 75.35% | 70.05% |
| Clarinet | 63.42% | 58.43% | 80.52% | 81.00% | 79.10% | 77.67% | 73.16% |
| Double bass | 89.60% | 82.99% | 96.22% | 95.27% | 96.03% | 96.22% | 93.01% |
| Flute | 43.28% | 38.88% | 62.59% | 57.46% | 54.52% | 52.57% | 51.59% |
| Horn | 25.12% | 31.53% | 12.32% | 14.78% | 14.29% | 19.70% | 21.18% |
| Oboe | 75.35% | 4.93% | 7.75% | 9.15% | 9.15% | 10.56% | 10.56% |
| Sax | 48.03% | 41.73% | 37.01% | 17.32% | 17.32% | 11.81% | 11.02% |
| Trombone | 83.48% | 71.23% | 90.88% | 86.89% | 81.77% | 75.21% | 74.36% |
| Trumpet | 74.02% | 25.49% | 39.71% | 40.69% | 42.65% | 39.71% | 42.16% |
| Tuba | 98.04% | 75.98% | 96.08% | 94.12% | 93.14% | 95.59% | 89.71% |
| Viola | 84.26% | 50.25% | 93.40% | 97.46% | 98.48% | 97.97% | 96.95% |
| Violin | 53.35% | 47.61% | 50.00% | 49.28% | 49.28% | 46.89% | 44.98% |

## 5.4 Projection Mechanism

Our final investigation contrasts the image quality and conditioning adherence of generated samples from two near identical diffusion models. The models differ only with regard to how they reduce the dimensionality of audio embeddings. The first model uses PCA to reduce the 1024-dimensional AudioCLIP embeddings into 256-dimensional representations, while the other incorporates a leaky ReLU activated fully connected layer. Both models employ 4 second text-aligned audio embeddings, aligning these findings with the previous study.

Findings indicate that the PCA variant falls short with regard to both evaluation criteria. Test FID scores for the PCA aligned model are substantially worse than the fully

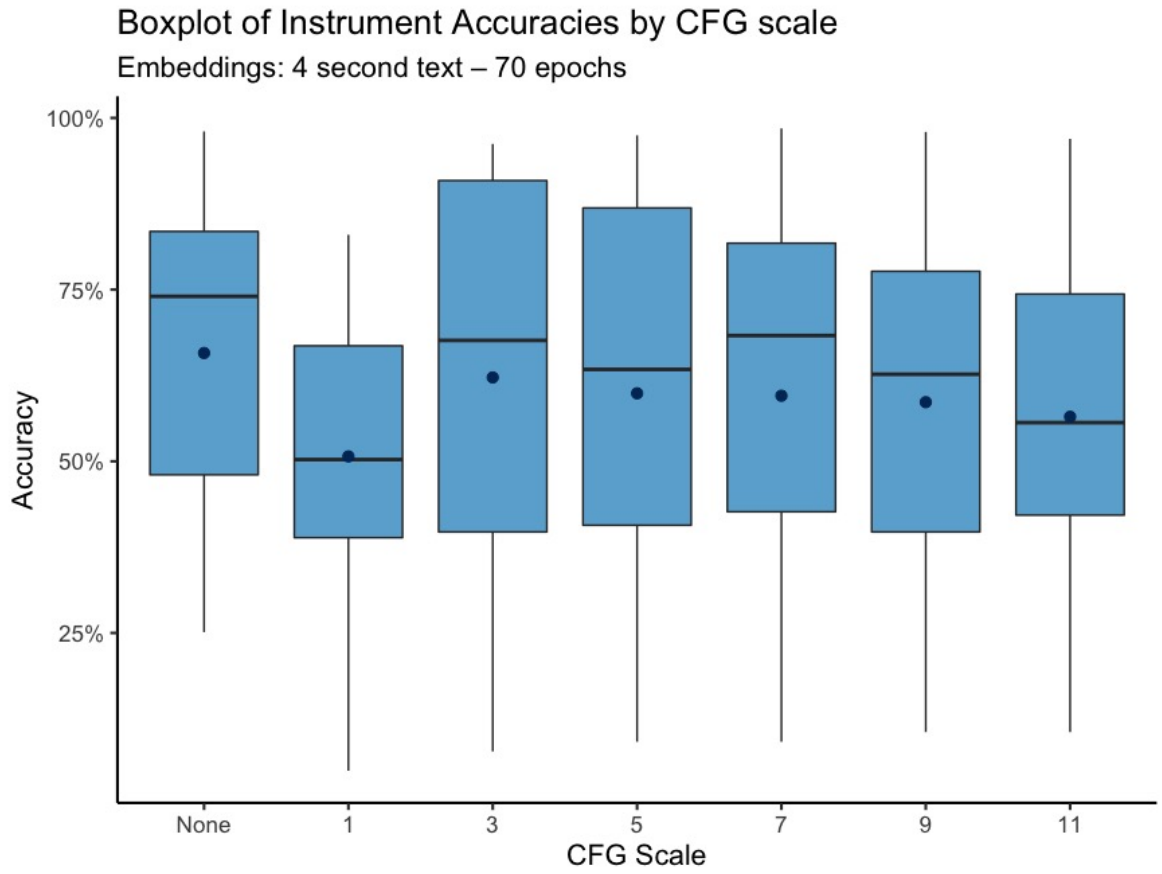



Figure 5.8: Boxplot of instrument classification accuracy by CFG scaling parameter.

connected variant. Specifically, the PCA aligned model returned a FID of 148.19, while the fully-connected variant achieved 78.69. Similarly, instrument classifier accuracy was lower in samples generated by the PCA model at 53.85% compared to the fully connected model's 68.66%, indicating worse conditional adherence.

Table 5.9: Performance comparison between projection mechanisms.

| Projection | FID | Accuracy |
|---|---|---|
| PCA | 148.19 | 53.85% |
| Fully Connected | 78.69 | 68.66% |

Visually, images generated by the PCA model showed a wider range of brightness, resulting in outputs varying from particularly dark to over-exposed (See Figure 5.9). On the other hand, images produced by the fully connected model displayed more consistent lighting, and visually aligned more closely with the original image distribution. As a consequence, we hypothesize that the discrepancies in FID scores could be a result of the differences in brightness distributions. We argue this because it is this aspect of PCA generated images that most substantially deviates from the distribution of testing images. In addition, we attribute this disparity to the lack of normalization components incorporated in the PCA variant. We infer this by observing that when dropout and layer normalization's were removed from the fully connected approach, images appeared similarly over-exposed. See Figure 5.10 for reference.

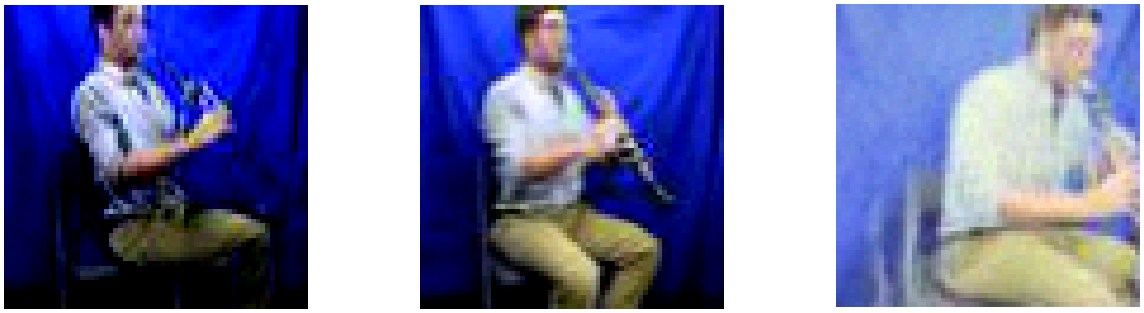
The improvement in conditioning adherence demonstrated by the fully connected model

is likely a result of the learnable nature of its transformation. Unlike PCA, the fully connected layer is optimized as a part of the image generation model. It is therefore capable of selectively extracting components of the original embedding relative to their significance to the image generation problem.

In contrast, PCA has no learnable component and necessarily reduces the dimensionality of the input while preserving as much information about the input as possible. In this case, PCA retains over 99.9% of the variation in the original embedding. Importantly, there is no guarantee that all of this information will be relevant to the image generation task. Thus, by simply retaining as much information as possible, PCA may inadvertently carry over a significant amount of irrelevant information into the image generation problem. Doing so would likely make it harder for the generative model to adhere to the pertinent aspects of the audio embedding, potentially explaining the reduction in conditional adherence.

(a) Dark (b) Normal (c) Light

Figure 5.9: Brightness variation in PCA model variant.



(a) Dark (b) Normal (c) Light

Figure 5.10: Brightness variation in non-normalized fully connected model variant.

## 5.5 Summary

In this chapter, we presented and discussed our findings from the experiments outlined in Chapter 4. This began with an initial investigation that focused upon three concurrently run experiments. Those being studies into embedding alignment strategies, optimal audio sample lengths, and appropriate training durations. We found that models conditioned upon text-aligned embeddings generally performed better with respect to conditional adherence and image quality. Those conditioned on image-aligned embeddings tended to perform slightly worse on both metrics. This said, their adherence to conditioning information ap-

peared to consistently improve as sample lengths were increased. Moreover, these improvements were not found to curtail at any point. As such, future research would do well to investigate this further.

We noted an interesting observation with regard to errors made by the generative model. While our best performing model did not adhere to conditioning information perfectly, aligning only 68.66% of the time, the mistakes it made appeared understandable. Namely confusing woodwinds with one another, as well as mistaking instruments constructed from brass. This suggests that the model appears to have learnt representations that resonate with our own understanding of these instruments.

With regard to training duration, we found that both conditional adherence and FID scores tended to plateau after, at most, 30 epochs. Negligible performance improvements were observed after this point. Interestingly, next to no overfitting was observed with additional training. We reason that this is likely a result of the high level of similarity between training and testing images.

Remarkably, our model performed better than state-of-the-art GAN alternatives in terms of FID scores. We reason that this is likely a result of the increased diversity in images samples present in diffusion models. That said, image resolution could also be a contributing factor, thus further work would do well to explore the effect increasing image size has on FID scores.

Our final two experiments investigated the effect of CFG and projection mechanisms respectively. The former study found that performance improvements were observed when CFG was incorporated. That said, these improvements were only realised when an appropriate scaling parameter $s$ was used. In our study we found $s = 3$ to produce the best results.

Lastly, our investigation into projection mechanisms found that using a fully connected layer to decrease audio embedding dimensionality was more effective than PCA. We attribute this to additional normalization components in the fully-connected approach, alongside its ability to selectively retain information from the original embedding, potentially helping remove redundant information.

# Chapter 6

# Conclusion

This research project focused on investigating the relatively under-explored machine learning problem of Audio-to-Image generation. We deemed this to be an interesting area, given its exploration of multi-sensory image generation. As humans, our understanding of the world around us comes from an integration of data from various sensory channels. When we merge this information, we gain a more comprehensive understanding than we would from relying on a single sense alone. This led us to consider the value of understanding how machine learning models bridge the gap between different sensory inputs. Audio-to-Image generation allows us to explore this question visually. Unfortunately, the domain is relatively under-explored with state-of-the-art models producing only low-resolution outputs on highly task specific problems [2][11].

To address this, we proposed the first integration of a diffusion image generator into the Audio-to-Image generation domain. We also drew inspiration from DALL-E 2 [14], additional incorporating an expressive audio encoding network based upon CLIP [17][6] into our generative framework. By integrating these architectural components, we hoped to improve Audio-to-Image generation performance given the success associated with their inclusion in the related field of Text-to-Image generation.

Our best performing model demonstrated a substantial improvement in image quality over existing GAN approaches, as measured in FID. Specifically achieving 75.32 on the metric, compared to 107.26 realised by the best performing GAN alternative [4]. It must be noted however that state-of-the-art GAN alternatives generate higher resolution images. As such future work will need to investigate how these models compare at comparable image scales. Unfortunately, our model was found to generate samples that aligned with conditioning information less frequently than state-of-the-art GANs. Our best model generated conditionally faithful images 68.66%, while the best performing GAN does so 89% of the time [4].

Our experimental analyses revealed a number of particularly interesting results. Namely, we found that conditioning upon text-aligned embeddings was more effective than image-aligned embeddings for our particular problem. We hypothesize that this may have been a result of the limited intra-class variance present in our dataset, and wonder if the same trend would be observed with a more diverse dataset. Additionally, we found models conditioned on image-aligned embeddings produced more conditionally faithful images given longer audio sample lengths. We reason that the additional information provided by longer audio samples could prove useful given the increased complexity inherent to image-alignment. Moreover, we found frozen image-alignment to yield more predictable results than mutual-alignment. We attribute this to the stable nature of the image-head in the frozen case.

In addition to the above, we also found that employing a learnable dimensionality reduction layer for audio embeddings was more effective than PCA. This effect was likely

a combined result of the additional normalization components incorporated in the former method, as well as its capacity to selectively incorporate information from the original embedding, depending on its relevance to the image generation task.

Finally, we found that incorporating classifier-free guidance was useful for improving model performance. Specifically, by setting the scaling parameter to 3, our model was able to achieve both better conditional adherence performance and FID scores. We also note that choosing the scaling parameter effectively has a significant impact on performance, thus highlighting the importance for future researchers to select this parameter careful.

Whilst our model faced challenges with respect to conditional adherence, the outcomes of this research project remain promising. Specifically they provide a solid foundation for justifying the inclusion of diffusion models into the Audio-to-Image generation domain.

## 6.1 Limitations

Whilst successful, our study was not without its limitations. We suffered from two major challenges in particular.

Firstly, we were required to work around quite serious computational constraints. While we had a substantial amount of resources at our disposal, the generative model required a considerable amount of time for training and evaluation. This issue restricted us from running multiple iterations of our model under identical settings, consequently limiting our ability to conduct statistical evaluations. Doing so is important given the degree of randomness inherent to the generative process. More extensive evaluations would have yielded more reliable results, enabling us to draw more definitive conclusions from our work.

Secondly, the dataset we employed only enabled a proof of concept project. Whilst popular within the literature, SubURMP lacks a sufficient amount of intraclass variation for complex analyses. Consequently, the task leaned more towards a class conditional diffusion problem. If a more diverse dataset was used, we could expect more intriguing results to be achieved, alongside the learning of richer latent data manifold which could be very interesting to explore. This said, the primary aim of this work was to demonstrate the feasibility of incorporating a diffusion model into the Audio-to-Image generation problem. As such we do not view this limitation as a major setback.

## 6.2 Future Work

Future work could build upon the current project by exploring a variety of promising directions. Firstly, we highly recommend that subsequent studies tackle the limitations we highlighted above. Doing so will likely result in more interesting and robust conclusions that could add significant value to the Audio-to-Image domain.

In addition there are also a number of additional paths to explore. One exciting prospect would be to incorporate a super-resolution network to increase image resolutions. These networks are commonly used in diffuse Text-to-Image generation tasks, enabling the upscaling of $64 \times 64 \times 3$ images to much higher resolutions. By doing so, one would enable a fairer comparison of image quality between state-of-the-art GANs that produce $256 \times 256 \times 3$ resolution images [4].

We also encourage future researchers to investigate different strategies for improving the conditional adherence of the network. We found this to be the biggest drawback of the current approach, thus solving this would be highly beneficial to the domain. Those interested could explore the effect longer sample lengths have on image-aligned embeddings, as our

findings showed this to be a potentially useful means of improving their performance. Additionally, researchers could also investigate the effect classifier guidance has on conditional adherence. Whilst this technique has not proven as effective in the Text-to-Image domain [14][12][13], it does not necessarily follow that the same is true for Audio-to-Image generation. Furthermore, researchers could also explore the effect training both sub-networks in an end-to-end fashion has. We found good results were achieved when the image generator could selectively chose what information to retain from the original audio embedding (i.e. with a fully connected layer). One could reasonably expect additional performance improvements to be achieved if the audio encoding network is trained with feedback from the image generation network.

Lastly, further research investigating the latent data manifold could prove interesting. We found interesting results when scaling CFG image representations and would welcome a more thorough exploration of this area. This could yield highly influential findings with respect to understanding how machine learning models blend information from both audio and image modalities. This is a large aspect of what makes the Audio-to-Image generation domain important, thus we strongly suggest future work analyse this in detail.

# Bibliography


[1] Y. Ding, F. Chen, Y. Zhao, Z. Wu, C. Zhang, and D. Wu, "A stacked multi-connection simple reducing net for brain tumor segmentation," *IEEE Access*, vol. 7, pp. 104011–104024, 2019.

[2] L. Chen, S. Srivastava, Z. Duan, and C. Xu, "Deep cross-modal audio-visual generation," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pp. 349–357, 2017.

[3] W. Hao, Z. Zhang, and H. Guan, "Cmcgan: A uniform framework for cross-modal visual-audio mutual generation," *arXiv preprint arXiv:1711.08102*, 2017.

[4] H. Chung, J. Shim, and J.-K. Kim, "Cross-modal contrastive representation learning for audio-to-image generation," *arXiv preprint arXiv:2207.12121*, 2022.

[5] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Esresnet: Environmental sound classification based on visual domain models," 2020.

[6] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Audioclip: Extending clip to image, text and audio," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 976–980, IEEE, 2022.

[7] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020.

[8] Z.-H. Zhou, *Machine learning*. Springer Nature, 2021.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[10] T. Jebara, *Machine learning: discriminative and generative*, vol. 755. Springer Science & Business Media, 2012.

[11] W. Hao, M. Han, S. Li, and F. Li, "An attention enhanced cross-modal image–sound mutual generation model for birds," *The Computer Journal*, vol. 65, no. 2, pp. 410–422, 2022.

[12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

[13] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *arXiv preprint arXiv:2205.11487*, 2022.

[14] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[15] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," 2022.

[16] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*, pp. 2256–2265, PMLR, 2015.

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[20] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, pp. 2048–2057, 2015.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[22] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International conference on machine learning*, pp. 1060–1069, PMLR, 2016.

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," 2016.

[24] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[25] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*, pp. 8162–8171, PMLR, 2021.

[26] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[27] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[28] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[29] H. Al-Tahan and Y. Mohsenzadeh, "Clar: Contrastive learning of auditory representations," in *International Conference on Artificial Intelligence and Statistics*, pp. 2530–2538, PMLR, 2021.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[31] 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," *Advances in neural information processing systems*, vol. 30, 2017.

[32] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are gans created equal? a large-scale study," 2018.

[33] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2020.

[34] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2019.

[35] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," 2021.

[36] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Esresne(x)t-fbsp: Learning robust time-frequency transformation of audio," 2021.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[38] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer.," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[39] M. Tao, H. Tang, F. Wu, X.-Y. Jing, B.-K. Bao, and C. Xu, "Df-gan: A simple and effective baseline for text-to-image synthesis," 2022.

[40] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 5907–5915, 2017.

[41] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," 2017.

[42] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 2017.

[43] A. Teolis, *Computational Signal Processing with Wavelets*, vol. 182. Springer, 1998.

[44] A. Guzhov, "Audioclip," *https://github.com/AndreyGuzhov/AudioCLIP*, 2021.

[45] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[46] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," 2021.

[47] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," 2019.

[48] T. Capelle, "Diffusion-models-pytorch," *https://github.com/tcapelle/Diffusion-Models-pytorch?organization=tcapelleorganization=tcapelle*, 2023.

[49] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications," *IEEE Transactions on Multimedia*, vol. 21, pp. 522–535, feb 2019.

[50] Y. Wen, R. Singh, and B. Raj, "Reconstructing faces from voices," *arXiv preprint arXiv:1905.10604*, 2019.

[51] A. Duarte, F. Roldan, M. Tubau, J. Escur, S. Pascual, A. Salvador, E. Mohedano, K. McGuinness, J. Torres, and X. G. i Nieto, "Wav2pix: Speech-conditioned face generation using generative adversarial networks," 2019.