NORTHWESTERN
UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS-2023-06

March, 2023

Adapting Latent Diffusion Models from Images to Class-Conditioned Audio Generation

Jackson Michaels

Abstract

Generative machine learning for waveforms is difficult due to the large number of time steps. 2-D frequency representations present a solution to this challenge. Diffusion models and latent diffusion models have been shown to exhibit extremely high quality image generation, we designed an LDM trained on spectrogram images with the purpose of generating audio with text conditioning. We have shown this training exhibits improved scoring against latent diffusion models trained on images.

Keywords

Machine Learning, Deep Learning, Audio Generation, Spectrogram Generation, Latent Diffusion Model, Transfer Learning

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The central goal of this work is to leverage deep image generation techniques to enable open domain audio generation, the first step is a proper understanding of image generation. Deep learning has a long history in generation with most of the focus being in image generation with models such as Generative Adversarial Models (GAN) [1, 2], Variational Autoencoders (VAE) [3], and more recently Diffusion Models [4, 5, 6, 7]. Image generation is a difficult task for many reasons but primarily the difficulty in determining a good metric for image quality has been a limiting factor in training deep models for generation. For this reason, Fréchet Inception Distance (FID) scores were created to function as a unifying metric for image generation. FID measures similarity between two image sets and has been shown to correlate highly with human judgment of visual quality and is a classic metric for image generation tasks [8]. FID scores are crucial in that they remove human judgment from the process but are still flawed due to the inherent subjectivity in judging generative models. Diffusion models are the newest type of generative model to perform state-of-the-art image generation and function differently from models such as GANs or VAEs. A diffusion model is a highly complex denoising model that is trained by adding noise to images and learning how to remove that noise. The next big development for diffusion models came with latent diffusion models [6], the difference being the addition of an autoencoder/decoder pair to reduce the complexity of the input space for the diffusion model. These models have been shown to be able to generate audio in the one dimensional time domain but not yet in the two dimensional frequency domain.

While audio generation is less explored than image generation there have been very successful models built for a wide range of tasks from text to speech to open-domain audio generation using a wide range of models similar to image generation [4, 9, 10, 11, 12]. These models fall into one of two categories based on the nature of their generation space. Firstly some models generate one dimensional waveforms directly and have shown impressive results though often have issues with

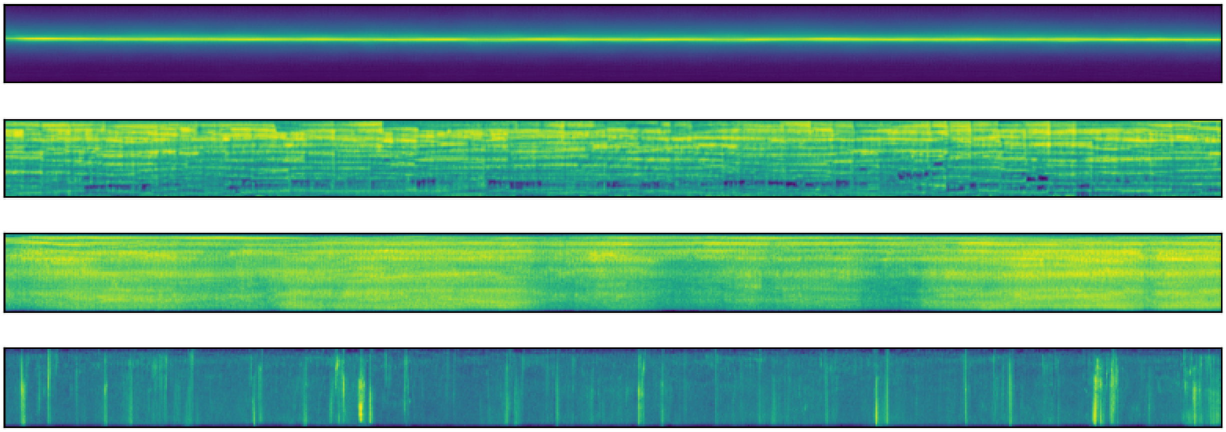

Figure 1.1: Example audio generation samples prompted with class guidance. 1st: sin wave. 2nd: music and song. 3rd: car engine idling. 4th: whip sounds.

the high sampling rate leading to lower temporal consistency across tens of thousands of time steps [4, 9, 11, 12]. The other approach is to generate a two dimensional spectrogram image and convert this to a waveform [10]. Figure 1.1 displays a cherry-picked sample of spectrogram generation. For example, the top image is a generated sin wave with a constant frequency, resulting in a single horizontal band of color. Reduced temporal resolution representations, such as spectrograms, can help with long-term temporal consistency but often require extensive processing to convert back to a waveform and often have a strong presence of artifacts. Figure 1.2 displays a comparison between these two representations.

A limiting factor for audio generation vs image generation is access to training data. Image datasets such as ImageNet [13] have been a fundamental resource in the development of powerful image generation models but accessibility for large audio datasets is more limited. Image generation models in general require extremely large amounts of training data when compared to other deep learning applications and would not be possible without datasets such as ImageNet and others. One dataset that is often used for audio generation is AudioSet, a dataset of over two million ten second audio clips that will be used for training in this paper [14]. While AudioSet is a crucial dataset for audio generation, the quality of labeling and size of the dataset are significantly reduced when compared to ImageNet.

Another important note on previous audio generation methods is their domain-specific genera-

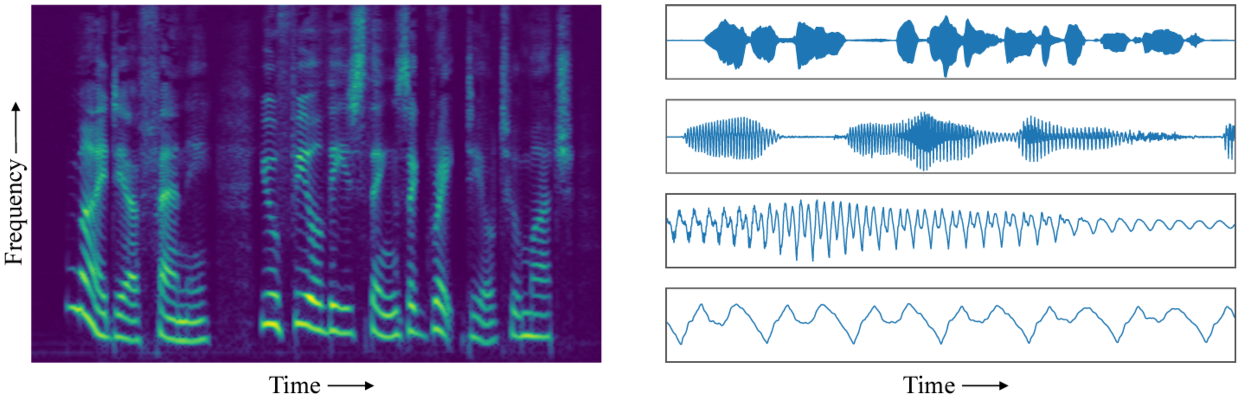

Figure 1.2: Comparison between spectrogram and waveform representations of the same four second audio clip. The waveforms are ranging from 1X to 125X magnification and span nearly 100,000 time-steps whereas the spectrogram spans roughly 400 [10].

tion. All of the aforementioned papers are domain-specific to speech, music, or similarly consistent domains. The goal of this work is to generate in an open domain space. This means there is little chance of surpassing these models in their domains but open the possibility to generate any audio sample required at a moment's notice. To assist in text conditioned generation and avoid mode collapse only a limited subset of the samples was used to ensure balanced classes resulting in only 21717 samples. Mode collapse is a term used to describe when a model learns to only output most common element of the distribution, as in the generation "collapses" to only generating the "mode" of the generation.

In this paper, I present a latent diffusion model (LDM) pre-trained on the LaionB image set [15] and fine-tuned with spectrograms generated from a balanced subset of AudioSet. I performed experimentation with various autoencoder architectures to find the best combination of autoencoder and diffusion model for spectrogram generation. As spectrograms have differences in structure compared to images I could not just use the model best suited for images and instead ran tests both training from scratch and fine-tuning autoencoders such as a masked autoencoder [16], vector quantized variational autoencoders (VQ-VAE), and KL-divergence variational autoencoders (KL-VAE). Finally, for sampling the generated images I use a lower resolution than the training set, 64 x 400 instead of 512 x 512, to closer reflect real spectrograms and show that this approach yields measurably improved results over 512 x 512 sampling.

The remainder of this thesis is structured as follows. In chapter 2, I will cover related works to this subject focusing on the history of deep generative models, audio generation, and diffusion models. In chapter 3, I will show the methods used for creating the model including detailing the dataset trained on, the selection of the encoder, the complete model, sampling methods, and results. In Chapter 4 I explore limitations both in hardware and data accessibility along with future research and experimentation to circumvent these issues. Finally, chapter 5 is a conclusive summary of my work.

CHAPTER 2
RELATED WORKS

While generative models are relatively recent, their history is fundamentally important to understand the task ahead. This work would not be possible without the extensive and impressive results from years of research which are fundamentally important to understand the goals and results of this work. To this end, chapter two will examine the important related works for LDM based audio generation starting with generative models in general, then audio generation, and finally diffusion models leading to latent diffusion models.

2.1 Deep Generative Modeling:

The core idea behind deep generative modeling is to train a model to generate new samples within the distribution of a training set, this used to be seen as intractable but recent advances have not only shown that generation is possible but can be very carefully controlled. The primary difficulty in deep image generation is defining loss and evaluation metrics as it is not clear what exactly successful generation means. Many generative models will use human-in-the-loop methods such as a mean opinion score or selection rates when judging one model against another. Comparing this against accuracy scores for classification models it is clear why there is difficulty in objective comparisons between generative models, to this end FID scores are common but not universally used to compare models. GANs were among the first models capable of high-quality image generation and continue to be an open field of study today [2]. A basic GAN consists of a generator and a discriminator where the generator receives a random noise vector as input and generates a sample in image space. Then the discriminator is trained to classify samples as from the training set or the generator. This approach leads to the generator learning to generate high-quality samples within the distribution of the training set or close enough to fool the discriminator. There continue to be developments on these models such as StyleGAN [1] which takes insights from the style transfer

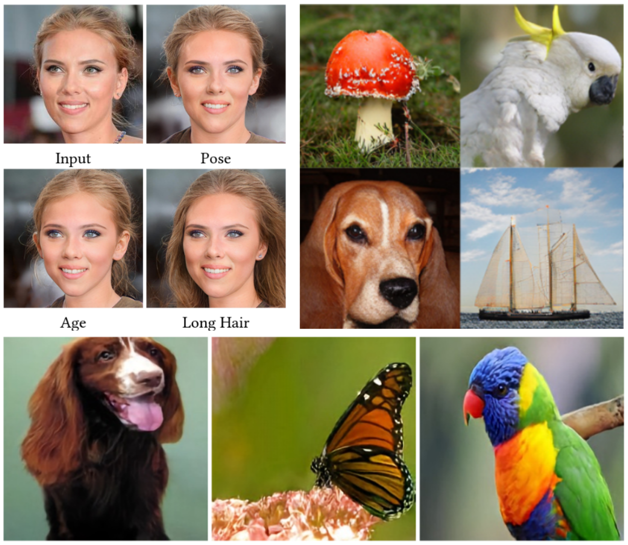

Figure 2.1: Comparing the generation from 3 state-of-the-art methods in deep image generation: Top left, StyleGAN showing editing of real-life images with conditioning. Top Right, transformer-based class conditioned generation. Bottom, VQ-VAE class conditioned image generation [1, 3, 17].

task to add more fine-tuned control to each stage of the convolutional generator network allowing for separation of high-level details (pose or identity) from low-level stochastic details (hair or freckles). Figure 2.1 shows an example of StyleGAN transforming an input image with pose, age, and hair adjustments. Another recent area of research for GANs has been into Vector Quantized GANs (VQ-GANs) which aim to leverage a learned code book to generate higher-quality images. A code book is a way of discretizing a continuous space. In a VAE these can be used by having a learned set of codes (a code book), then when an image is encoded instead of the latent vector being passed to the decoder, the code vector with the smallest euclidean distance to the latent vector is passed instead. GANs have limitations however, they can be difficult to train due to the adversarial nature of their model, commonly suffer from mode collapse, and show a lack of diversity [3].

VAEs are another well explored method for generative deep learning that have important differences when compared to GANs. in [3], the authors make this distinction explicit by categorizing generative models into two main types: likelihood or probability-based models, including VAEs, flow-based models, autoregressive models, and diffusion models; and implicit models such as GANs. The aforementioned distinction is important for judging the performance of GANs against likelihood-based models due to the fundamental differences in the mathematical foundation of how they generate. Due to this and the fact that probability-based models set the probability of all samples in the training set to be 100%, these models cover all modes of data and do not suffer from mode collapse [3]. Another important separating factor is the presence of auxiliary models. Which for this purpose are defined as models used only for training and not used in inference, for example, the discriminator in a GAN or the encoder in a VAE. Mode collapse, posterior collapse, or training instability stemming from the joint training of two networks are common points of failure for VAE and GAN-based approaches that diffusion models can avoid [11].

Transformers address some of these issues but primarily the aforementioned issue around the inductive bias, that adjacent pixels are related, built into these previous models they gained through the use of convolutional layers. In contrast to this, transformer models do not have a built-in bias and in turn, can learn more complex relationships within the input beyond simple adjacency [17]. This is not without issues however as these inductive biases are included for a reason, generally speaking, the aforementioned bias is correct. Transformer-based models can learn more complex representations but as a result, they also must learn all relationships within the data, including that pixels near each other should be related [17]. This results in the increased expressivity of transformer models coming as a trade-off for exponentially increased computation costs. [17] address this issue by combining convolutional and transformer models into a single unified generative pipeline. Their general approach is to use a convolutional VQ-GAN to generate a code book and then use an autoregressive transformer to model the composition of the context rich parts found in the code-book. They state this approach not only helps alleviate the strong locality bias of CNNs but helps to address the spatial invariance that can be a result of using shared weights across every position [17].

2.2 Deep Audio Generation:

Audio generation has a long history, in some ways longer than image generation; more than fifty years ago Hiller Jr & Isaacson created the first music generator based on Markov chains [12]. For years these music generation approaches were rule driven and required explicit rule sets to generate. This is useful for interpretability but very weak when it comes to generalizability and diversity. Modern data-driven methods have been shown to alleviate these issues leading to WaveNet, the current benchmark model referenced by most audio generation papers [11]. WaveNet is a probabilistic autoregressive model that conditions each sample on the previous samples and generates raw waveforms. Generally speaking, audio generation techniques fall into one of two domains, two-dimensional frequency (such as spectrograms) and one-dimensional time domain (such as a waveform). A spectrogram is a two dimensional representation of sound waves over time where the X-axis represents time, the Y-axis represents frequency, and the intensity at the point represents the intensity of that frequency at that point in time as seen in Figure 1.2. Spectrograms are additive and can easily be overlaid with one another allowing for easy mixing of audio samples. Spectrograms do a good job of capturing the audio in a visual structure but do lose some features, primarily the phase of the sound waves is lost in the conversion. The largest difference between these two approaches is the size of a time-step. Waveform-based approaches have to handle the fact that a single second of audio can span tens of thousands of time-steps pressing the limit of long-term consistency [10]. This is very apparent in WaveNet as they state their models' receptive field extends back about 300 milliseconds, only allowing the model to "remember" the previous 2-3 phonemes when generating the next sample [11]. On the other hand, generation in the frequency domain has addressed this issue due to the far lower temporal resolution required for generating a sample with 400 time steps vs. 160,000. A typical spectrogram could have only 200 time steps for a 10 second audio clip allowing for far better long-term temporal consistency as a model only needs to ”remember" up to 200 previous events to be fully informed. [10] introduces MelNet and display this difference in an experiment asking human judges to pick between a MelNet and a WaveNet sample for a longer-term structure. MelNet achieved 100 percent of the evaluators picked MelNet generations across two of the three samples and 95.8 percent in the third showing how large a difference generation in the frequency domain can make when focusing on long-term temporal consistency [10]. One limitation of spectrogram-based generation is the final conversion to an audio wave. MelNet addresses this issue with a multiscale generation procedure where they first generate a low resolution spectrogram and progressively up-sample it to high resolutions until it has enough detail to be converted to a waveform without excessive artifacts and without needing to directly work in the time domain [10].

Generating in the frequency domain is not the only method for long-term consistency in audio generation. Methods such as Jukebox from Google handle this issue with a multi scale VQ-VAE to compress the audio into discrete codes and then model those using an autoregressive transformer [9]. They show that this configuration is capable of generating audio with coherence for up to multiple minutes without losing sound quality. This approach is similar to the one employed by [17] where they combine the expressiveness and lack of bias in transformers with the reliability and presence of bias in a VQ-VAE. These approaches differ slightly however as Taming-Transformers uses a VQ-GAN and Jukebox uses a VQ-VAE but the intuition is the same. Jukebox employs another approach that was seen in MelNet with varying levels of detail in its model. They first train three separate VQ-VAE models with different temporal resolutions and later can sample from any of these code books. This allows varying levels of abstraction with larger windows driving higher temporal consistency and smaller windows assisting in audio quality. They then trained up-sampling models to allow for conversion between these representations. For instance, modeling the probability of a middle resolution sample given a low resolution sample or a high resolution given middle and low. A 3-level approach to detail assists in the final generation by allowing low resolution generation, which is better suited for longer temporal consistency, then up-sampling to a higher resolution spectrogram to assist in the conversion back to audio. This is necessary as the temporal resolution of a waveform is enormous when compared to a spectrogram, so up-sampling can artificially increase the temporal resolution of a spectrogram enough to assist in the transformation.

2.3 Diffusion Models:

Diffusion models are a relatively new probability-based generation model primarily used for image generation. They function by progressively adding gaussian noise to an image as a forward pass and then learning how to remove this noise by learning to predict the highest probability noise pattern in backward passes conditioned with a text prompt. Once the model can predict the noise pattern it can remove said noise bit by bit in a number of steps to generate natural images from pure noise. Most diffusion models use a U-Net model architecture to learn the denoising process as shown in Figure 2.2. While currently GANs hold the state-of-the-art in most image generation tasks when comparing quality metrics such as FID scores this comparison isn't perfect. FID scores fail to capture the diversity in generation the models are capable of [5]. As stated previously GANs suffer from a few common issues: difficulty in training due to the dual training of multiple models as well as mode collapse without very careful selection of hyperparameters and regularization [5]. While GANs are still state-of-the-art these issues cause them to scale poorly and make applying them to new domains more difficult. These issues were the driving force in developing likelihood-based models that can compete with GANs in image quality leading to diffusion models. These models initially had issues competing with GANs in image quality though they did exhibit far more image diversity and are easier to train and scale. However, generally speaking sampling from likelihood models, with the notable exception of VAEs, is slower than sampling from a GAN [5].

Latent diffusion models [6] made the next large innovation in diffusion-based generation by adding an encoder-decoder pair to the pipeline as seen in Figure 2.3. Instead of functioning in raw pixel space the LDMs first pass images through a powerful pre-trained image encoder to a lower dimensionality latent space. One of the largest drawbacks of diffusion models is their extremely long training time requirements often using hundreds of GPU days, however this can be alleviated by first lowering the size of the input space with an autoencoder. These LDMs can achieve new state-of-the-art scores for tasks such as inpainting and class conditioned image synthesis while also achieving extremely competitive scores for tasks such as text-to-image generation, unconditional

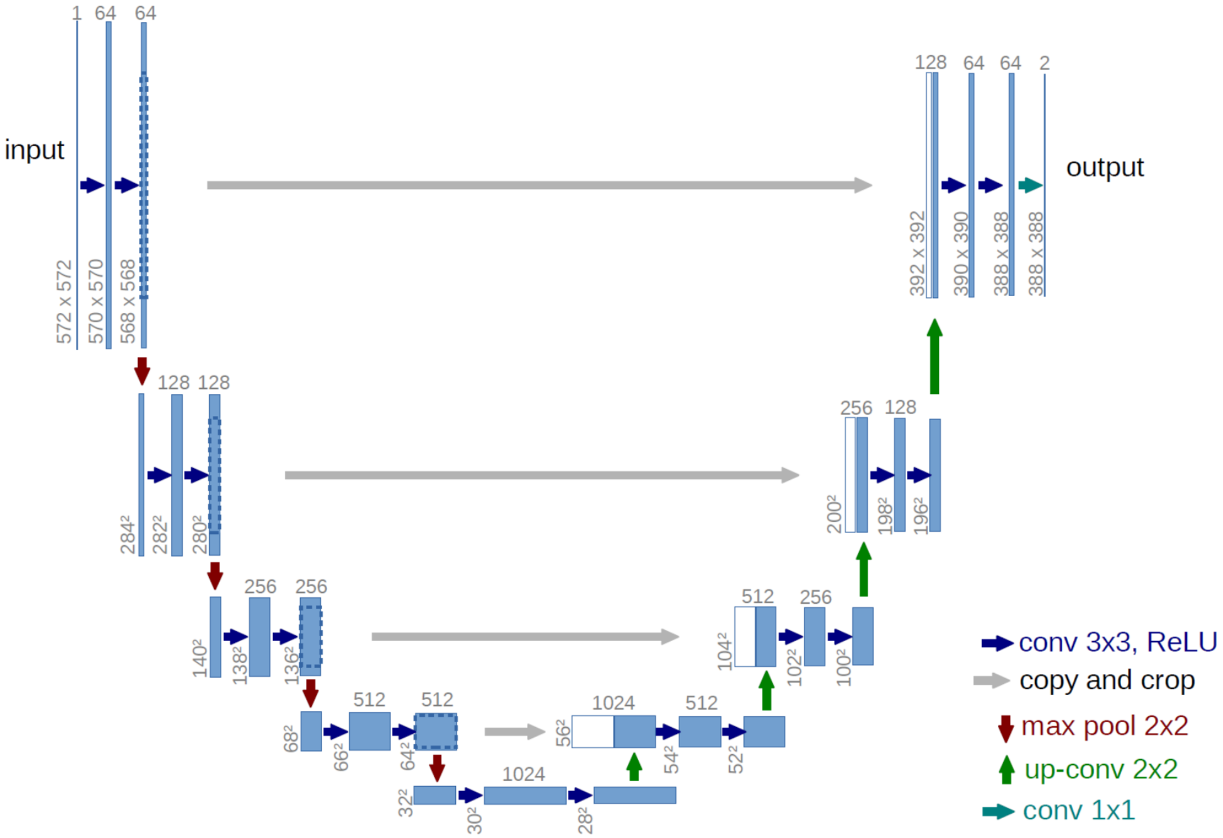

Figure 2.2: detailed model diagram of standard U-Net architecture. This specific model is for image segmentation but the structure is the same. U-Nets use a combination of convolutional layers to extract rich image features and then recombine these with earlier representations via the "copy and crop" connections also known as skip-connections [18].

image generation, and super resolution.

All of this is achieved while still being significantly easier to compute than traditional pixel spaced diffusion models [6]. The models released by the authors[1] form the basis for the audio generation pipeline I created. These models were chosen as a base due to the lack of diffusion-based audio generation in the frequency domain and their impressive results on image generation tasks.

Audio generation is not completely unseen with diffusion models. DiffWave is a diffusion-based audio generation technique that generates directly into the one-dimensional time domain and does not employ the latent space trick used by LDMs [4]. This leaves a lot of room for

[1] https://huggingface.co/CompVis/stable-diffusion-v1-4

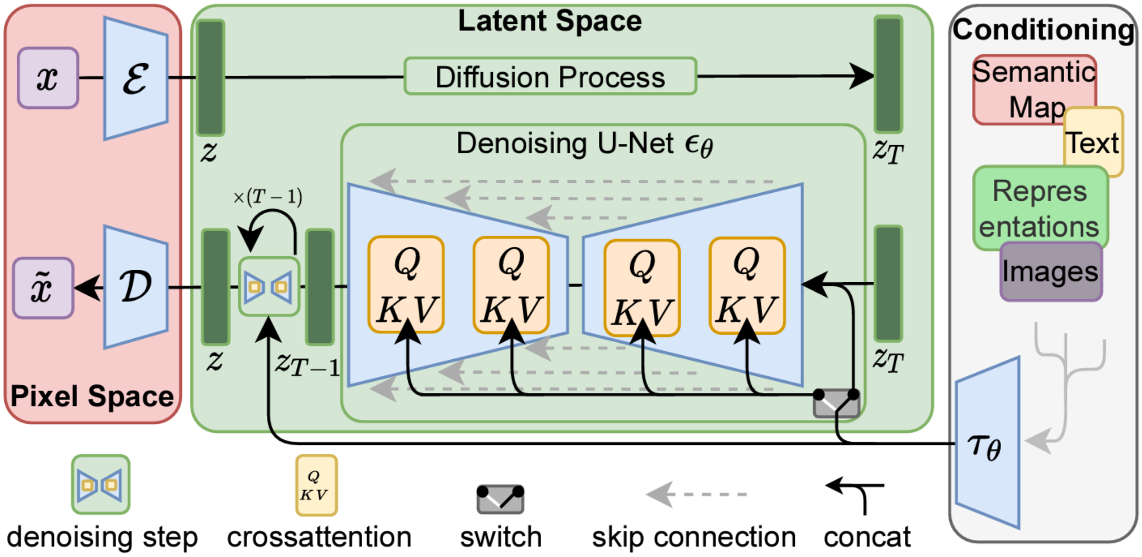

Figure 2.3: Model diagram for latent diffusion model displaying encoder decoder pair reducing input complexity for denoising U-Net [6].

potential improvements and tradeoffs when compared to our method of latent space generation in the frequency domain. DiffWave is most comparable to WaveNet when comparing the trade offs between diffusion and GANs for audio generation and exceeds or meets WaveNet in most common audio generation tasks. As there exists no universal comparison metric, DiffWave uses mean opinion scores. On a neural vocoding task DiffWave was able to match WaveNet while exceeding them in unconditional generation by a large margin [4].

CHAPTER 3
METHODS

With the context for this work established the next piece is an exploration of the practical methods I used when conducting this work. The dataset selection was not a simple choice as there are important trade-offs between increasing the size of the training set and reducing class imbalance. Once the data was selected, I explored how to construct the pieces of the LDM that would eventually generate spectrograms beginning with exploring the encoder/decoder stage of the LDM pipeline. With these pieces in place, the next step is to fine-tune the LDM with my chosen dataset and encoder/decoder pair. Finally, I discovered the ideal generation technique for this model and compare its results to the pre-trained generator.

3.1 Dataset:

AudioSet is one of the largest and most used datasets for short labeled audio clips and was first created to bridge the gap between labeled image datasets, such as ImageNet, and labeled audio datasets. AudioSet consists of 2,084,320 ten second wav files extracted from YouTube videos and was hand-labeled into 632 audio event classes [14]. These classes are organized into a hierarchical structure designed to be unambiguous and mutually exclusive between the classes the first two levels of which are shown in Figure 3.1. For this reason, nonexclusive classes such as “dog sounds” and “bark” are simplified to have “bark” be a child of the “dog sounds” label. To further improve the clarity of labels and assist in the hand labeling process many samples “back up” the hierarchy when a label cannot be determined. For instance, AudioSet has classes for “growl”, “bark”, and “howl” which are all ideally distinct, however, in an audio sample these can easily be confused. When this is encountered the labeler would instead assign “dog sounds” as the label leading to a less specific but ideally more accurate labeling scheme. This hierarchy is what I leveraged to allow for more verbose labeling in the training set.

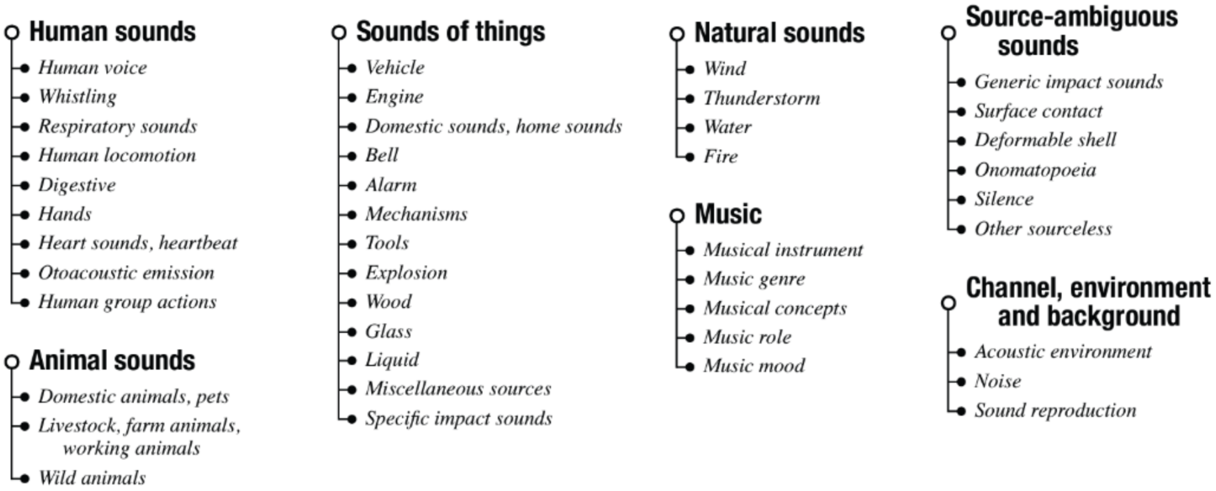

Figure 3.1: First two layers of AudioSet labeling ontology displaying the range of audio domains included in AudioSet [14]. The open-domain nature of AudioSet is the main benefit employed to improve non-speech or music-based audio generation.

To assist in text conditioned generation and avoid mode collapse only a limited subset of the samples was used to ensure balanced classes resulting in only 21717 samples. Without this balance, speech and music would make up 50% of the dataset and the model could learn to only generate speech or music. This is common with GANs as the generator can learn that a single image always fools the discriminator, and once this is achieved there is no incentive for the generator to generate anything else. Another cause of mode collapse is due to an imbalance in data. For instance, if a classifier model is trained on a dataset of 98% class A and 2% class B, it could achieve 98% accuracy by only predicting A. The same intuition is what motivated me to use a balanced subset of AudioSet to avoid this collapse as AudioSet is nearly 60% speech and I wanted the model to generate any sample in its training set, not just speech.

These raw soundwaves are first transformed into 128 x 128 pixel spectrogram images using torchaudio's built-in spectrogram converter[1]. As the LDM was fine-tuned initially on square 512 pixel images, these 128 pixel images were up-sampled using a built-in PyTorch up-sampling layer to 512 pixel resolution. This size and shape were selected to most closely match the pre-training of the diffusion model to assist in learning the structure of a spectrogram without needing to re-train

[1] https://pytorch.org/audio/stable/generated/torchaudio.transforms.MelSpectrogram.html

lower level features. for generation the pipeline is prompted with a string matching the structure of the labels closely starting with “a spectrogram of the sound of” to reliably reflect the training data and an output resolution of 64 x 400. This output resolution was chosen for two reasons, firstly it assists in the final conversion back into a waveform and secondly, it results in better samples when compared to the ground truth spectrograms created directly from AudioSet with FID scores. After generation, to convert back to an audio wave the Griffin-Lim algorithm, an optimization-based phase solver, is employed as an inversion of the initial waveform-to-spectrogram transformation [19]. One common issue with audio generation in the two dimensional frequency domain is the final conversion back to a one dimensional time domain, the difficulty comes from the loss of phase information about the various frequencies. Griffin-Lim attempts to solve this issue by randomly initializing phase estimates and alternates between forward and inverse passes to find the ideal set of phases for a given spectrogram. As a result, even when processing directly from sound wave to spectrogram and back to sound wave there will be noticeable artifacts that are present in generated samples due to the nature of the transformation.

To address the large data requirements for training some samples have been augmented and provided new labels based on the hierarchical nature of audiosets classifications. For example to supplement the dataset an additional 5000 samples were created by randomly selecting two samples with the same parent classification and then averaged together. For these synthetic samples instead of labeling with the same scheme as the other samples these new spectrograms were given the caption “a mel spectrogram of the sound of multiple overlapping [parent class name]s, including [class name one] and [class name two]”. The intuition behind this was not only to provide more samples for the dataset without largely unbalancing the classes but also to provide examples of overlapping sounds ideally to assist in the generation of more dynamic scenes.

3.2 Initial Testing:

The first stage in an LDM pipeline is the use of an autoencoder to lower the complexity for the U-Net and assist in learning which is fundamentally important to the improvements latent diffusion has over traditional diffusion models. The U-Net itself is very rarely modified from its initial presentation as any changes only reduce its performance. The first stage however is highly customizable as it is only required to simplify the input space for the U-Net. When first developing this model a number of encoder architectures were tested, namely a Masked Autoencoder (MAE), a KL-VAE, and a VQ-VAE. The KL encoder and VQ encoders were both presented as options in [6] and had been shown to perform well for this purpose informing my final decision to use a KL-VAE after comparing the models. One of the initial goals of this research was to determine the efficacy of MAEs for this purpose due to their success in audio encoding and even reconstruction after masking [16].

MAEs appeared to be a promising direction as they had not been used for this purpose before and have shown improved performance in reconstruction over other autoencoder models. The other goal behind using MAEs was the potential for their performance on reconstruction to assist the LDM in generating accurate samples as almost an additional generative step.

After some testing however, I found that MAEs had a fundamental issue for this purpose in that the latent space had to be larger than the input space due to encoding of patch level metadata such as class labeling and positional information used in the reconstruction shown in Figure 3.2. This issue can be resolved by using fewer patches with a larger stride from the original image but in the end, a simpler model was chosen both to more accurately reflect the training environment of

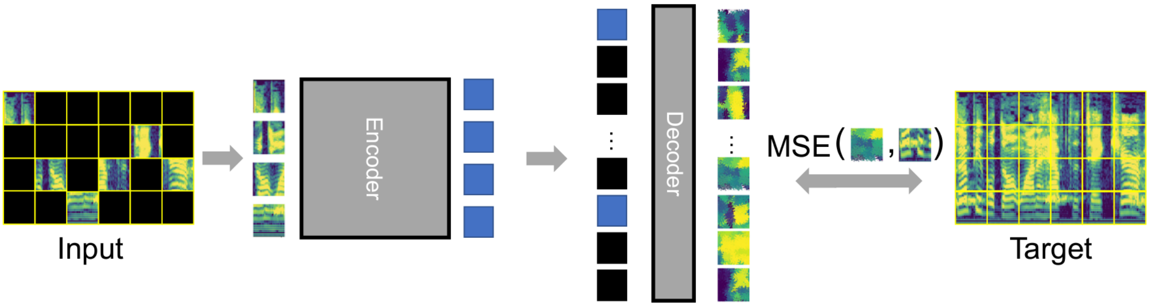

Figure 3.2: Model architecture for Masked Autoencoder importantly showing patch level reconstructive abilities. Not shown is the size of the latent space which results in increased complexity over the input space [16].

the models pre-training and reduce the possible losses from large gaps in the input from the wider patch stride.

The next model used for testing was a VQ-VAE pre-trained on LaionB data [15] data, a 5,85 billion image dataset with text captioning. In testing, I found that the pre-trained model failed to capture the importance of high frequency details in reconstruction and resulting in an overly smooth image.

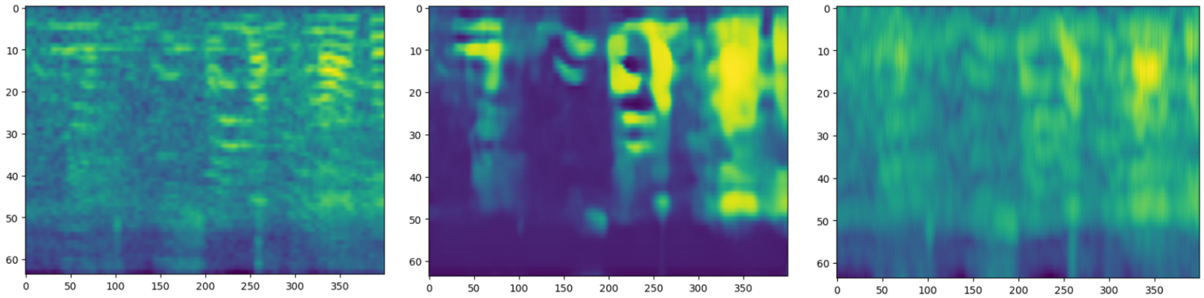

Figure 3.3: Example of fine-tuning improvements on VQ-VAE model. Left: input image. Middle: pre-trained output. Right: fine-tuned output. While the fine-tuning substantially improved the reconstruction quality and high frequency detail, after fine-tuning the model still fails to capture the complexity of the input image and appears overly smoothed.

As shown in Figure 3.3, fine-tuning the encoder-decoder pair is required to conserve the higher frequency information in darker regions of the image, however, the final resulting image still lacks in detail and has an overly smooth appearance when compared to the input. In the input image there are clear sets of parallel frequency bands that are very important for correct sounding audio, whereas the reconstructions blur them into single large patches which may be similar in an image but for a spectrogram exhibit entirely different audio features. For this reason the final model, a KL-VAE was selected for testing next.

There are a couple of advantages to transitioning to this approach: Firstly the availability of pre-trained models is substantially better for the KL-VAE than for the VQ-VAE meaning less time would be needed for training this first stage. Secondly, the image reconstruction was substantially improved when compared against the VQ-VAE as shown when comparing the reconstructions from Figure 3.3 and Figure 3.4. In summary, the MAE was powerful but the size of the latent

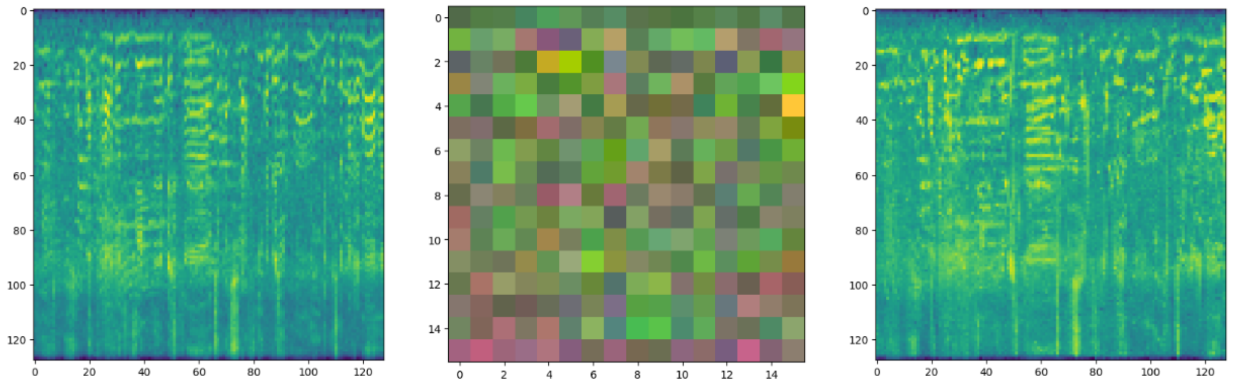

Figure 3.4: Encoding performance of KL-VAE showing. Left: input image. Middle: latent representation of the image. Right: reconstruction. while this is substantially improved over VQ-VAE the output images still lose some details, especially in the top right region.

space defeated the purpose of a VAE stage, the VQ-VAE showed promise but ultimately failed to conserve high frequency detail even with fine-tuned training, and finally the KL-VAE avoids all of the aforementioned issues. For these reasons the final model employs a KL-encoder/decoder pair, however, there is potential for future work with these other two models.

3.3 Latent Diffusion Model:

This model follows the work done by the Latent Diffusion group and uses their pipeline architecture with fine-tuned models. The basic structure of this pipeline has three components: a VAE, the U-Net generative model, and a pre-trained text encoder. The encoder stage consists of a KL-VAE with four layers encoding from an input space of size 512 x 512 to a latent space of size 128 x 128 yielding a 16 times reduction in dimensionality. This encoder is only required for training and during inference, the encoder stage is not used as the noise vector passed to the U-Net can be generated in latent space directly. A classic U-Net also consists of an encoder stage and a decoder stage of sorts, both of which are composed of ResNet blocks. They function very similarly to a VAE with the encoder reducing the dimensionality while extracting important image features while the decoder returns these latent representations to the higher resolution input space (for the U-Net) ideally having removed the noise. Specifically the decoder side is trained to predict the noise residual which is used to compute the new denoised image. The U-Net has another important feature

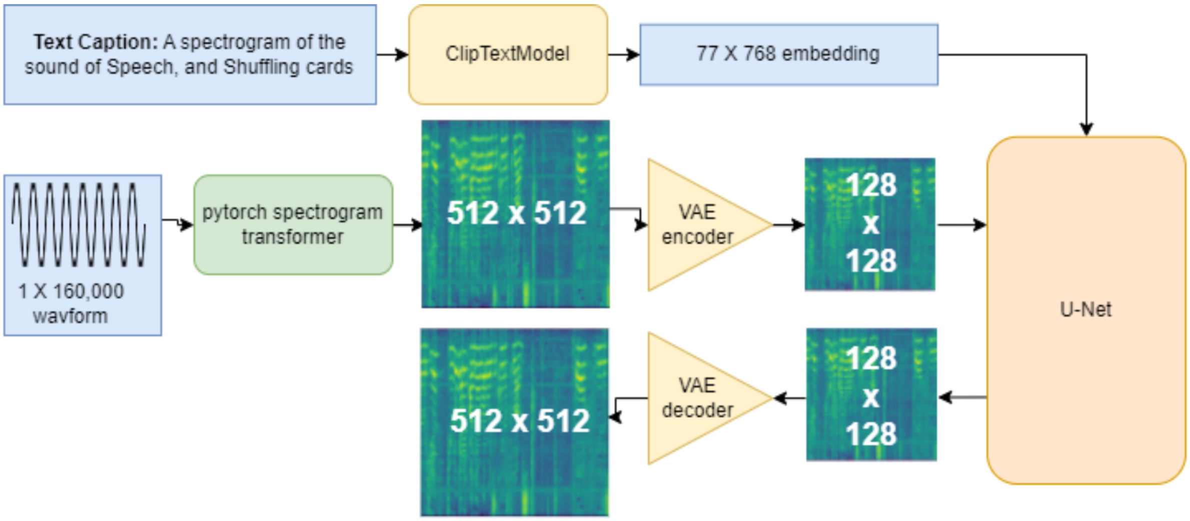

Figure 3.5: Model diagram of training and fine-tuning process. Soundwaves are first transformed into 2D spectrograms, then fed into the VAE encoder stage, passed through the U-Net where cross attention based transformer blocks combine the image data with the text embeddings, the output image is then passed through the VAE decoder stage, and the final loss is calculated by reconstruction loss between the input and output 512 x 512 images.

in the skip-connections between the layers of the U-Net allowing for the decoder model to have access to relevant spatial context when reconstructing. Without this, the decoder stage would only have the feature dense latent representation to reconstruct which is lacking in critical spatial feature information. For instance, the convolutional layers may successfully extract the features that certain shapes or patterns are present but not how they are located in relation to one another.

Finally, the text conditioning is applied through the use of cross-attention layers added to both the encoder and decoder sides in attention blocks to inform both feature extraction and image reconstruction. For the training of this model, text conditioning is applied by first passing the text descriptions through a CLIP text embedder encoding them to a 77 x 768 embedding size and then to the cross-attention layers. For this model the text encoder is not fine-tuned and instead an off-the-shelf model called ClipTextModel was used for text encoding [6].

This pipeline is built off of a pre-trained LDM initially pre-trained on LaionB datasets consisting of images paired with short text descriptions. The training process for diffusion models often consists of multiple training stages on different image sets with different resolutions. The model used in these experiments was first trained on a 256 x 256 resolution set from Laion2B, then a set of 512 x 512 images from the Laion-high-resolution. This checkpoint was further trained on a set called Laion-improved-aesthetics, a subset of Laion2B with an estimated aesthetics score greater than 5.0. Finally that model was fine-tuned with the Laion-aesthetics v2 dataset with a 10% drop rate on the text conditioning to improve their classifier-free guidance sampling, for a total of 150,000 hours of training spread across 32 machines each containing 8 A100 GPUs. This is the extent of the pre-training and the model was additionally trained on the dataset of spectrogram images from AudioSet. Fine-tuning was done on a single NVIDIA 3090 GPU with 25000 optimization steps over the course of twelve hours.

3.4 Generation:

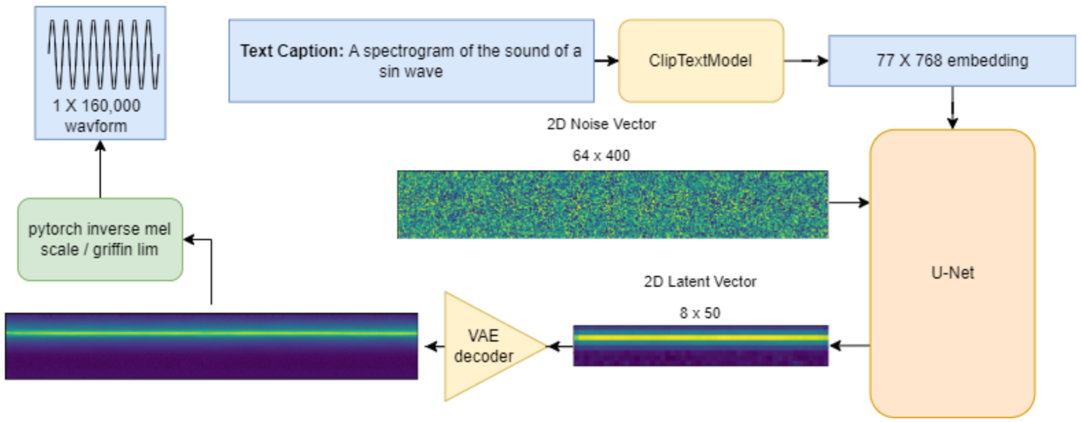

Figure 3.6: Model diagram of generation process including text conditioning and latent representations. This model can generate at any resolution evenly divisible by 8 due to the convolutional layers. This is why 64 x 400 generation is possible despite 512 x 512 images being used for training.

Generating samples using an LDM is similar to training with the notable difference in the image input. For sampling instead of feeding in an image you instead feed an image of pure noise of arbitrary shape and the model outputs an image of the same shape as is shown in Figure 3.6. Initially the LDM was prompted to generate new samples with a resolution of 512 x 512 as this is the size and shape of the fine-tuning data provided. However through testing when generating at such a high resolution the resulting images often have repeating patterns in the vertical frequency dimension as seen in Figure 3.7. These artifacts lead to worse audio reconstruction and lower FID scores when compared to an alternative generation size. I discovered that when the model is prompted to generate at 64 x 400, a far more typical shape for a spectrogram, the resulting image completely removed the repeating patterns and resulted in an image with far higher similarity to reference spectrograms from AudioSet directly.

3.5 Results:

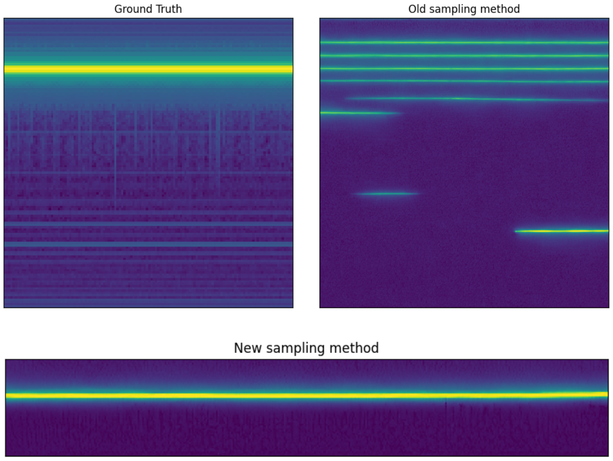

Figure 3.7: Comparison between generation techniques for a sin wave. As shown in the ground truth this spectrogram should feature a single flat horizontal band across the entire image. Left: ground truth from AudioSet. Right: 512 x 512 generation. Bottom: 64 x 400 generation. The old sampling method has problematic vertical repeating patterns that do not resemble the ground truth data whereas 64 x 400 removes these artifacts.

With this fine-tuning the model can generate an image with a very low resemblance to the natural images it was pre-trained on and with the appearance of a spectrogram however, it failed to capture much of the important details for accurate sound wave reconstruction. Figure 3.7 shows examples of the generator getting close to but not successfully modeling the ground truth images. As a result, some metrics used for audio generation such as mean opinion score and other human-in-the-loop methods will have trouble with this current model. However, with improvements in training and access to judges mean opinion scores would be a valid metric for evaluation. For this reason, FID scores are used to measure accuracy and consistency with the training set as is common in audio generation evaluation.

Model	Resolution (px)	FID Score
Pre-Train	512 x 512	183.63
Pre-Train	64 x 400	271.93
Fine-Tuned	512 x 512	140.00
Fine-Tuned	64 x 400	**72.86**

Table 3.1: FID score and sampling time compared between old and new sampling methods showing a substantial decrease in FID score. The bolded text shows the best result.

During experimentation, I found that sampling at this 64 x 400 resolution improved performance significantly both qualitatively in comparing generated samples ground truth samples and quantitatively in FID scores as shown in Table 3.1. While these samples are far from high-quality when converted back to audio the images show specific signs of learning class specific structural details as shown in Figure 3.8. The speech example from the aforementioned figure shows clear discrete words with breaks between them as would be expected from a person speaking. Similarly, this can be seen with the dog barking as well. Compare this to the music sample which fails to capture the specific details in music but does capture the presence of a “beat” or repeating sound roughly evenly spaced throughout the piece. These results show clear signs that the model is learning the required structures and more importantly is learning to diversify its generation.

The process of getting to this point consisted of many smaller steps. Firstly selecting a balanced subset from AudioSet to encourage balanced generation and avoid mode collapse. Secondly, I tested a number of VAE architectures including an MAE, VQ-VAE, and finally selected a KL-VAE for the final pipeline. Next, I integrated this encoder stage into the LDM model and fine-tuned the diffusion model for spectrogram generation via text conditioning. Finally, I experimented with various generation methods for the final results eventually selecting 64 x 400 images which achieved the lowest FID scores among all experiments I conducted.

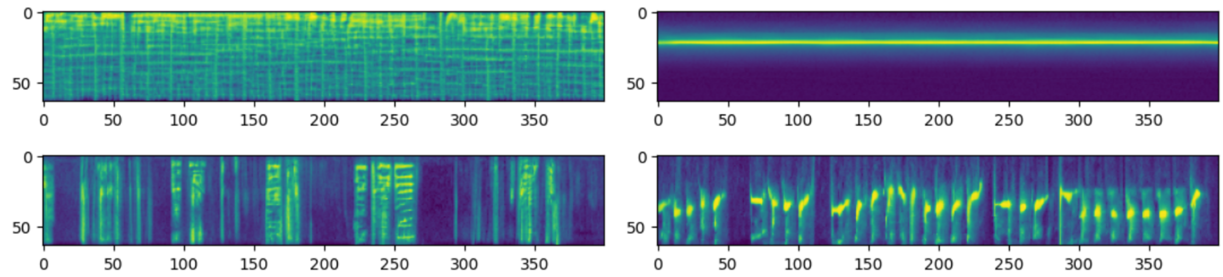

Figure 3.8: Four generated samples from the final model with the classes of music (top left), a sin wave (top right), speech (bottom left), and a dog barking (bottom right). They show important structural differences between the classes such as discrete audio events in the case of a dog barking or the constant and repeating nature of music.

CHAPTER 4

DISCUSSION

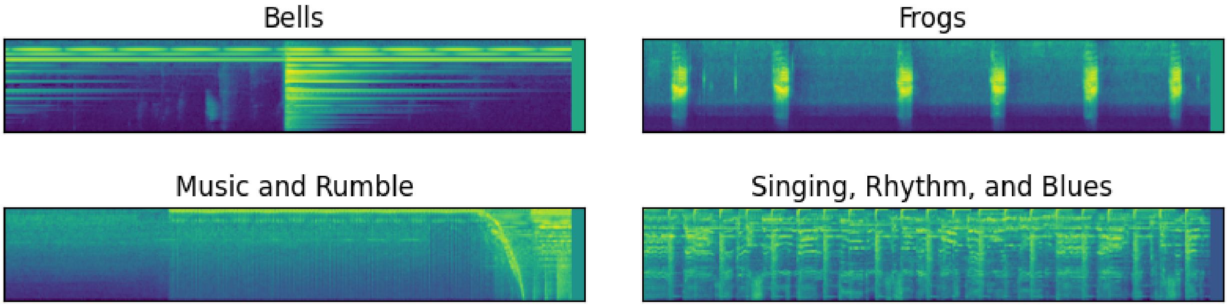

Figure 4.1: Display of the variation in open domain audio samples between four classes ranging across domains. An important distinction is a change in structure, especially between the frog sample and the bell sample.

While the goal of the process was to generate accurate sounding audio in a wide range of domains, from speech to mechanical sounds, this proved to be a more difficult task than initially thought. Domain-specific audio generation has received extensive research, particularly in music and speech which was a motivating factor for our exploration of wider domain generation based on the range of classes in AudioSet. This task has been shown to have significant difficulty when compared to speech or music generation due to its diversity, lack of consistent structure and quality of ground truth data due to weak labeling [20]. Figure 4.1 shows a sample of this variance. While natural images also contain an extremely wide range of variance, the highly sensitive nature of spectrograms to any flaws in their construction leads to this being a more difficult task than open domain image generation. For instance, if an image generation model produces an image with invalid shading the image is still recognizable. However, if a spectrogram has high intensity regions that should not appear the entire audio clip can be affected and result in unrecognizable noise.

Some specific directions for future work could help address these issues and allow for higher fidelity open domain generation. Firstly this model was trained on a balanced subset of AudioSet only including ≈1.30% of the total dataset to assist in a more balanced class conditioned gener-

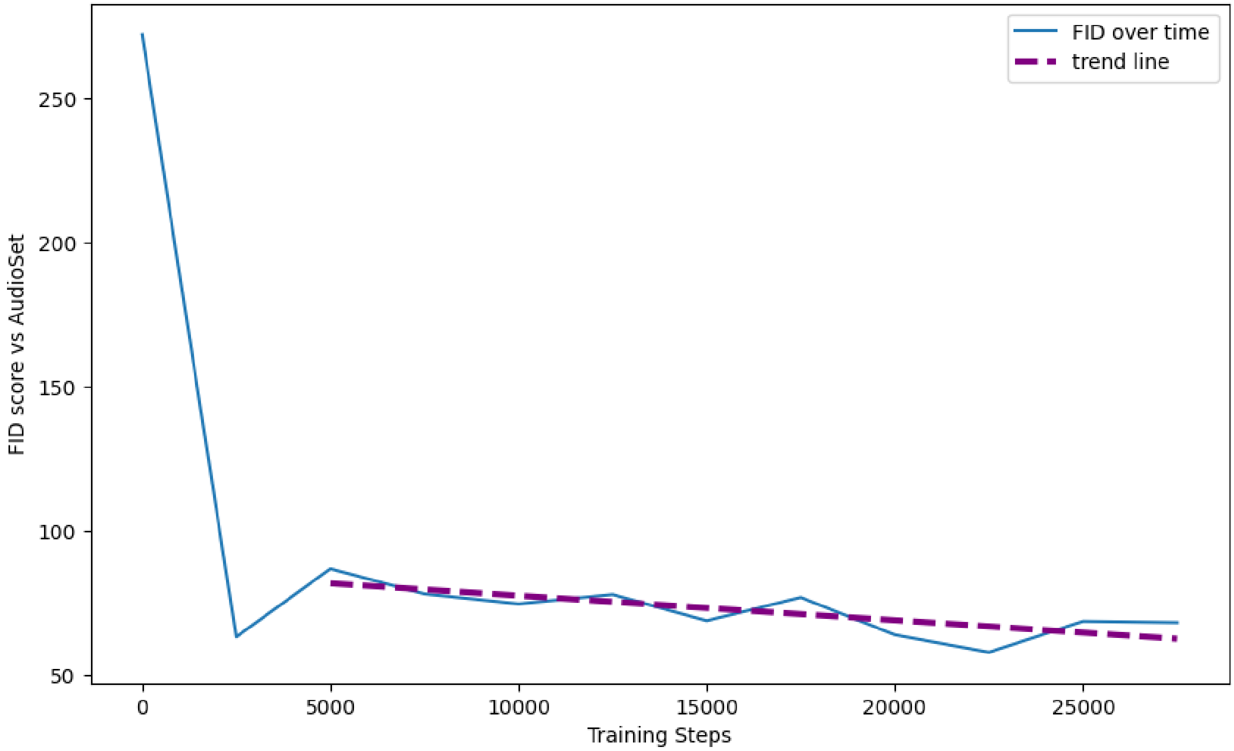

Figure 4.2: FID scores calculated based on 500 image sets generated by evenly spaced checkpoint models in fine-tuning process with a trend line after the first 2 checkpoints to avoid initial instability in the trend line.

ation. This was done to direct the model more strongly with the labels and avoid mode collapse due to the massive imbalance in classes. The inclusion of far more images could have assisted in general spectrogram generation and avoiding artifacts, though would weaken the open-domain motivations behind this model; however, I did not conduct this experiment and left it as a direction for future research. Another area for improvement is in hardware limitations, this model was initially trained for 150,000 hours on an AWS cluster across 32 machines each with 8 A100 GPUs where as the fine-tuning ran for only 12 hours on a single GPU machine. While this is substantially shorter training than the initial pre-training period it should be sufficient for fine-tuning purposes [6]. These estimates are made for fine-tuning with natural images and it is possible that with the resources the Stable Diffusion group had I would be able to continue training this model enough for truly accurate audio reconstruction. Figure 4.2 shows that when FID scores are calculated for checkpoint models in the fine-tuning process there is a slight downward trend that implies with more extensive training on more powerful hardware would continue to improve these results.

Another promising direction to improve this pipeline would be through more specific and powerful methods for the spectrogram-to-audio transformation. Many other papers use pre-trained transformers for this, I was not able to use these as they are domain-specific but it's possible an open domain data-driven transformer could improve audio fidelity substantially. Despite these limitations, the results achieved are quite interesting and show that with further training and more data LDM generation of spectrogram-based audio is possible.

CHAPTER 5
CONCLUSION

To conclude, in this paper I present a fine-tuned LDM pre-trained on the LaionB image dataset with training continued on a balanced subset of AudioSet. This model was able to generate images with a $\approx$50% decrease in FID score when compared to the pre-trained version and generate accurate samples for the simpler audio classes such as sin waves and other flat tones as shown in 3.7. This model is trained by feeding 512 x 512 pixel spectrograms with a text description into a KL-VAE and ClipTextEmbedder respectively. these are then fed into a U-Net for denoising with cross attention between the CLIP embedded text descriptions and the transformer layers of the U-Net. Finally, the output is fed through the decoder stage of the KL-VAE and converted back into waveforms with the Griffin-Lim algorithm.

As the next steps to improve and continue this work, there are two categories for improvements. Firstly address the hardware/data limitations I faced in creating this pipeline by acquiring more machines for training and weakening the balance of the AudioSet to include more samples. Secondly, there are several interesting applications and experiments for future work, primarily in out/in painting with spectrograms and training a more powerful spectrogram-to-waveform transformer model. With successful out painting a user could create arbitrarily long and complex audio sequences when combined with in painting the user could even mask specific sections or even frequencies and have the model regenerate these patches with text conditioned control. My findings through this research are exciting and lead to even more possibilities for future works to expand and improve on the work done here.

REFERENCES

[1] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[2] I. Goodfellow *et al.*, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[3] A. Razavi, A. Van den Oord, and O. Vinyals, “Generating diverse high-fidelity images with vq-vae-2,” *Advances in neural information processing systems*, vol. 32, 2019.

[4] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “Diffwave: A versatile diffusion model for audio synthesis,” *arXiv preprint arXiv:2009.09761*, 2020.

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

[6] 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*, 2022, pp. 10 684–10 695.

[7] A. Q. Nichol *et al.*, “Glide: Towards photorealistic image generation and editing with text-guided diffusion models,” in *International Conference on Machine Learning*, PMLR, 2022, pp. 16 784–16 804.

[8] 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.

[9] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.

[10] S. Vasquez and M. Lewis, “Melnet: A generative model for audio in the frequency domain,” *arXiv e-prints*, arXiv–1906, 2019.

[11] A. van den Oord *et al.*, “Wavenet: A generative model for raw audio,” in *9th ISCA Speech Synthesis Workshop*, pp. 125–125.

[12] l. a. hiller jr. and l. m. isaacson, “Musical composition with a high-speed digital computer,” *journal of the audio engineering society*, vol. 6, no. 3, pp. 154–160, 1958.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[14] J. F. Gemmeke *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2017, pp. 776–780.

[15] C. Schuhmann *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[16] P.-Y. Huang *et al.*, "Masked autoencoders that listen," in *Advances in Neural Information Processing Systems*.

[17] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 873–12 883.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[19] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.

[20] X. Liu, T. Iqbal, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, "Conditional sound generation using neural discrete time-frequency representation learning," in *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2021, pp. 1–6.