

SuperiEAR: Diffusion Methods for Audio-Denoising

Luke Mann
Stanford University
lukemann@stanford.edu

Moritz Stephan
Stanford University
moritzst@stanford.edu

William Denton
Stanford University
wdenton@stanford.edu

February 10, 2023

1 Abstract

Oftentimes in digital audio communications, there are perturbations which interfere with the end user’s understanding of the human voice. There are many existing solutions to filter static noise that has statistical patterns, but there is a need for solutions that can fully remove background sounds, such as cars honking, heavy machinery, or explosions. We created a diffusion-based machine learning model tasked with minimizing these artifacts to improve the intelligibility of voices leveraging recent strides in the realm of diffusion-based models. Our main contributions are showing that such a model produces qualitatively viable results when operating on raw waveform data as well as a novel way to construct arbitrarily large audio-denoising datasets with minimal effort. All our dataset generation, modeling, and visualization code is available on GitHub.¹

2 Introduction

Audio artifacts are prevalent in a multitude of audio sources and can cause miscommunication and inconvenience. Similarly, visual artifacts can be prevalent in image generation models. Recent diffusion models have effectively removed noise in generating imagery. [1] Thus, we sought to apply those same concepts of diffusion models to audio data. For training our model, we used .wav files which contained audio of people speaking with added artifacts, such as sounds of a dog barking or car

horns. Using the original audio and the inserted artifacts, we trained this model to produce noise-free output. When evaluating the model, the input was an audio .wav file with artifacts, and the output was a .wav file with those noises removed via the inference process of diffusion. Our model can be applied to several circumstances, including but not limited to clearing up athletic event communication, augmenting ear-piece software for the hard of hearing in busy environments (such as a bar or nightclub), and helping workers to understand each other in busy industrial environments for better safety and efficient communication. In high-pressure situations, such as law enforcement, 911 calls, or military operations, removing not just static noise but also background sounds would have a clear positive impact. There has been plenty of prior work on enhancing voice signals in audio using statistical methods and basic filtering. More advanced research by groups such as Facebook AI Research using encoder-decoder based model architectures [2] and Pascual et. al [3] using Generative Adversarial Network (GANs) have demonstrated that deep networks outperform traditional methods at audio denoising. Our approach differs in that it uses a diffusion probabilistic model to denoise the raw waveform.

3 Related Work

There has been extensive previous work done working with audio datasets to improve the intelligibility of human speech. Most of this work has focused on more traditional models, such as several models which used regular or deep Auto-Encoders (AE) such as Lu et. al and Raj

¹<https://github.com/austrian-code-wizard/SuperiEAR>

et. al. [4] [5] Our motivation in exploring diffusion is that diffusion yielded improvements over AEs and VAEs in the domain of images. Wiener filtering, such as described in the work done by Chen et. al [6] and Le Roux et. al. [7] is a popular non-ML approach. Wiener filters, however, require strong assumptions about the nature and attributes of the interfering noise, specifically, they require observing the interfering noise without the noise of interest. The process uses the audio signals of the noisy artifacts to strain out the interfering noise in the messy "interested" signal, or the audio signal that we are interested in denoising. Previous diffusion-based denoising models are the last type of model relevant to the methods we explore. Of particular note is the DiffSound paper by Yang et. al. [8] and DiffWave from Kong et. al [9]. Building off of the SOTA work by Ho et. al., [10] DiffSound uses diffusion with text embeddings to generate audio from text. Although this application is different to the application we sought to focus on, the generated audio is extremely promising for applying diffusion to audio. Furthermore, given the promising results generated by diffusion in the realm of imagery such as by Rombach et. al [1] in eliminating noise from imagery, we sought to similarly apply this process to audio. The novelty of our approach is not only in our application but also in our data input. Existing diffusion approaches to sound apply the learning process and inference to the spectrogram of the audio, while we are directly applying diffusion to the waveform representation of the audio. This approach yielded novel results explored in the next sections.

4 Dataset and Features

We created a completely novel dataset that contains snippets of clear human-voice audio as well as a collection of different noise types. For the human-voice audio, we collected 34 audio clips from YouTube videos ranging from historical documentary movies to academic lectures. Altogether, these clips with clear voices accounted for dozens of hours of audio and over 1G of data. We then segmented the audio tracks into chunks containing only voices using PyDub's Db-level-based silence detection and truncated all chunks to 7s length, resulting in over 2000 training samples. Finally, we scraped 14 short clips containing realistic noise from fireworks, busy streets, in-

dustrial machinery and more. For each of our 2000 samples, we picked a random noise clip and a random starting point in the first 3.5s of the clip before overlaying the noise with our sample and then only used the latter half of the clip to ensure there was noise in the entire sample. We aimed to avoid overfitting in specific sound patterns by varying where in the clip a noise starts, but if overfitting issues occur in the future, there are many possibilities to utilize audio distortions for data augmentation. All our final clear and noisy samples were normalized to a sampling rate of 16000 Hz stereo audio, which we transform into a mono signal by taking the mean across the two channels.

5 Methods

5.1 DCAE Baseline

As a first baseline, we used a Deep Convolutional Autoencoder (DCAE) to project the raw noisy waveform into a latent space and then recover the de-noised sample using a decoder network. We modeled our baseline after the state of the art by Defossez et. al [2] which demonstrated lower objective loss than previous DNN and GAN-based architectures. Our DCAE uses 5 encoder layers and 5 decoder layers. Each layer has consist of a convolution layer with kernel size 3, an instance normalization layer, and a ReLU activation. The five encoder convolution layers have output channels of size 1, 128, 128, 256, and 256, respectively. The six decoder convolution layers have output channels of size 256, 256, 256, 128, 128, and 1, respectively. In the final decoder layer, the output is a single channel with a Tanh activation since the raw waveform is in the range $[-1, 1]$.

Similar to Defossez et. al [2], during training and validation we measure loss as the mean square error of the outputted audio compared to the sum of the spectral reconstruction loss and the L1 loss.

The spectral reconstruction loss is the sum of the spectral convergence loss and the magnitude loss such that:

$$J(y, \hat{y}) = \|y - \hat{y}\|_1 + \sum_{i=1}^n J_{stft}^{(i)}(y, \hat{y})$$

$$J_{stft}(y, \hat{y}) = L_{sc}(y, \hat{y}) + L_{mag}(y, \hat{y})$$

$$J_{sc}(y, \hat{y}) = \frac{\| |STFT(y)| - |STFT(\hat{y})| \|_F}{\| |STFT(y)| \|_F}$$

$$J_{mag}(y, \hat{y}) = \frac{1}{T} \|\log |STFT(y)| - \log |STFT(\hat{y})|\|_1$$

5.2 PCA and Latent Space Investigation

Using the latent space of the DCAE models, we investigated how the data representation differs between the limited encoded form and the raw waveform. To do so, we employed PCA on the original waveform of the noisy data and the DCAE embedding of the noisy audio.

5.3 Spectral Gate Baseline

As a non-machine learning baseline, we use a popular Python package implementing the Spectral Gate technique for audio denoising. The algorithm applies FFT and then computes statistics over each frequency band. It then computes a smoothed mask using the statistics and subtracts it from the signal to obtain a denoised signal. [11]

5.4 Diffusion Model

There has been plenty of work using diffusion models in the audio space such as for upsampling using NU-Wave by Han et. al [12]. The main contribution to using diffusion models for denoising audio is DiffWave from Kong et. al [9], but this implementation operates not on the 1D raw waveform but on 2D data generated by applying a Fast Fourier Transform (FFT) to the waveform to generate signals for different frequency bands. We modified the DiffWave architecture to operate on the raw waveform.

Diffusion is a process in which noise is introduced to data in discrete timesteps, which represents a Markov chain without learnable parameters. At each step of the chain, the model predicts the noise signal that was just introduced, effectively learning to separate random noise from data. In our case, the data we start with is the clear audio. At each timestep, we condition on the audio data with the artificial noise introduced during dataset generation. During inference, we start with random noise and iteratively reverse the diffusion process conditioned on a noisy audio

file. After fully reversing the diffusion, the model should have transformed the random noise into a clear version of the noisy audio we conditioned on. Kong et. al applies a trick to reduce the number of diffusion steps by scaling the inference noise schedule, which we use as well. While in general, it is intractable to calculate the distribution of clean samples given a sample with added noise, Ho et. al formulates a tractable version of the ELBO, which gives us a loss function that we can optimize instead.

$$J(\theta) = \mathbb{E} [\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t, x_n)\|_2^2] \quad (1)$$

Here, $\epsilon \sim \mathcal{N}(0, I)$ is Gaussian noise that we sample for each training step, $t \in [0, T]$ is the diffusion step, $\bar{\alpha}_t$ is the cumulative noise level at the corresponding diffusion step t , x_n is the audio that we want to denoise, and x_0 is the clear audio. [10] For the model architecture, we use the non-autoregressive bidirectional dilated convnet proposed in Kong et. al [9] and modify it to accommodate raw waveform data by changing the number input and output channels to 1. We use a dilation cycle length of 10 as recommended by Kong et. al but modify the number of residual channels from 64 to 48 and the number of residual layers from 30 to 24 to accommodate our smaller GPU memory. [9]

5.5 Evaluation Metrics

While we subjectively evaluate the output according to clarity, the sound quality of the voice, and success at removing artificial background noise, inspired by Kong. et al, we created an objective metric to measure the quality of the generated audio. We utilize the tiny version of OpenAI’s pre-trained Whisper model for Automatic Speech Recognition (ASR). We feed the output audio samples of all our methods along with the clear audio and the samples with injected artificial background noise through the ASR model to get an NLP transcription of the words in the track. We then compute the word error rate (WER) between the clear audio transcription compared to the transcriptions of all model outputs. While this is an objective output, it is not perfect since it is biased by the ASR model we use and model-generated audio might lower the performance of ASR models significantly compared to natural audio, even if it sounds good to humans. Further, the

word error rate is sensitive to even single-character differences in words. With S as the number of word substitutions, D as the number of word deletions, I as the number of word insertions and N as the number of words in the target text, the WER is defined as

$$WER = \frac{S + D + I}{N} \quad (2)$$

6 Results

6.1 Latent Space Analysis

We found PCA of the DAE embedding had a larger presence and variance distribution around the mean of the waveform PCA. We hypothesise that this could signify a more nuanced representation of the most frequent sounds, ie people speaking. Furthermore, the PCA of the embedding shows few to no outlying points, which could explain why the AE is so effective in eliminating outlying noises, ie in denoising, as those high variance points of the original waveform could be understood as the unexpected non-human noises in the noisy audio.

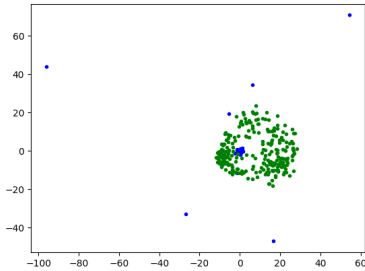
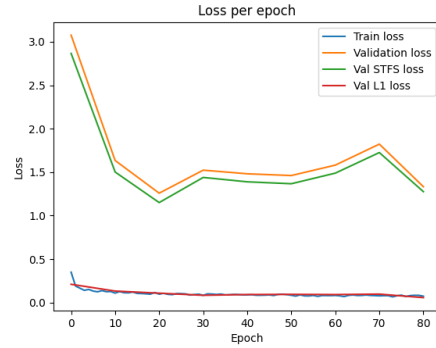


Figure 1: PCA of waveform and AE embedding of noisy audio in 2D. The waveform consists of blue data points whereas the embedding is represented in green.

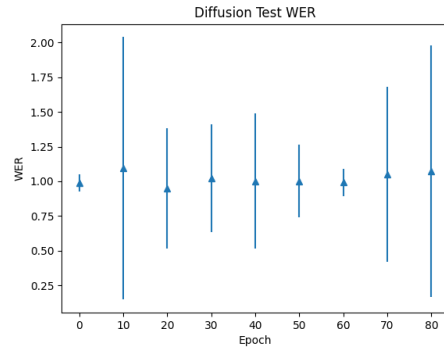
6.2 Model Analysis

We trained the diffusion model for 10 hours, which equaled 80 epochs, and were able to obtain outputs that resemble human voices and are closely aligned with the desired clean audio, albeit with significant static noise. We utilized a batch size of 10, a learning rate of $2 * 10^{-4}$, and no weight regularization since DiffWave does not utilize it either. The resulting model is successful, however,

at removing the injected background noises. The training loss of the model successfully converges. The validation loss decreases and might decrease even more with longer training. For validation, we used the same L1 loss that is used for training but additionally, we also compute the spectral reconstruction loss (STFS), which gives us an indicator of how closely aligned with the clear audio our output is, both in terms of frequency bands and respective amplitudes.

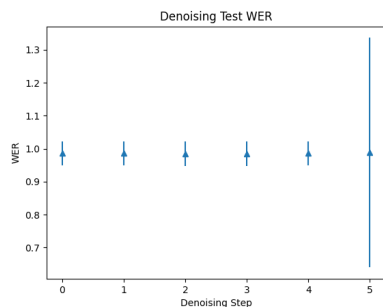


We also compute the word error rates on the validation set of the diffusion model at different training epochs. The word error rate on average always stays around 1 which can be attributed to issues with the evaluation metric itself since it results in a WER of 1 for outputs with only one word, which often happens due to remaining static noise. Based on a qualitative evaluation of validation output samples, we determined that the checkpoint at epoch 40 was the best performing. In the chart below, this could be explained by the mid-sized variance at that epoch which indicates that there was a significant number of good outputs.

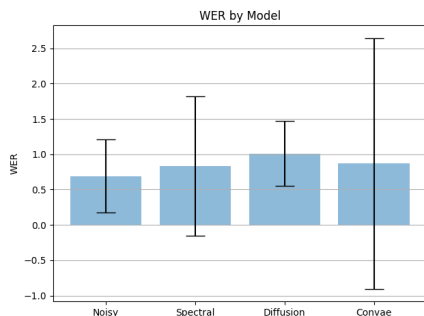


We can also look at the WER for the diffusion model

outputs after a variable number of reverse diffusion steps to highlight the gradual improvement in quality. Again, due to issues with the WER, the average is always around 1 but we can see that the variance increases significantly at the last iteration, indicating that there are many more good outputs.



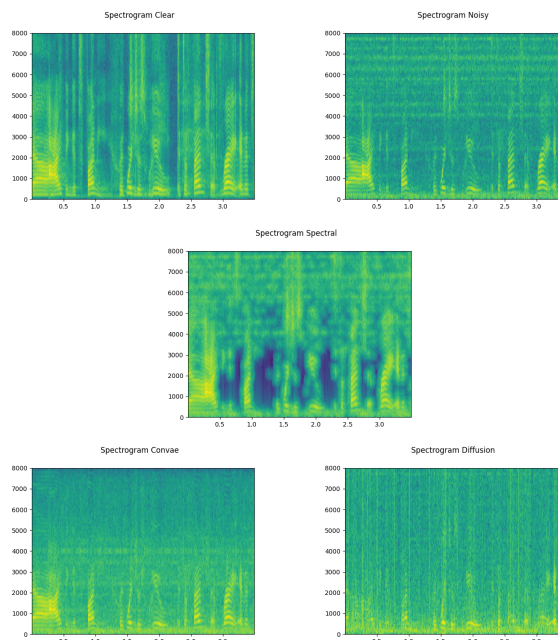
Finally, we can compare the WER of all our different approaches. We see that simply based on this objective metric, all models perform worse than ASR on the audio with the injected background noise itself. However, we believe this is due to issues with the ASR model on computer-generated inputs. Qualitatively, the output of the DCAE model sounds the most natural and successfully removes the injected noises, which the diffusion model successfully does too. The spectral gate baseline results in mainly unnatural-sounding audio that is poor in quality.



In the code repository for this paper, we include several test audio outputs for all models as well as clear and noisy examples.

Another qualitative metric we investigated is comparing the Mel Spectrograms of the different model outputs. It is visible that all 3 approaches compute output that looks similar to the clear audio but one can also recognize the

static noise present in the diffusion output. Based on this evaluation, the DCAE output looks most closely aligned with the desired output.



7 Conclusion and Future Work

We were able to demonstrate the potential for using diffusion models in audio denoising and were the first to do so on the raw waveform. Further, we created a framework for the simple generation of audio-denoising datasets. Still, there is significant potential to improve the final results. Parallelized training on multiple GPUs for a longer time on a larger dataset would certainly be useful since our validation loss was still decreasing when ending training and Kong et. al report training for about 1200 epochs to obtain non-noisy audio. Further, we would want to experiment with different residual layer sizes as well as a learning rate scheduler to prevent overshooting at later stages in the training process. To get a better understanding of the performance of raw-waveform approaches, a direct comparison to DiffWave on the same dataset would be useful.

8 Contributions

Will Denton was focused on writing the code to scrape the audio files as well as defining what kind of audio we should use for both the voice clips and the noise. In addition, he investigated implementing diffusion models with sound and the PCA of the AE embedding and raw waveforms. Moritz Stephan built the data preprocessing pipeline, worked on training the diffusion model, and researched evaluation methods. Luke Mann focused on the AE baseline model and researched approaches for the diffusion model. Finally, all members contributed to the report and graphics.

References

- [1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” 2021. [1](#), [2](#)
- [2] A. Defossez, G. Synnaeve, and Y. Adi, “Real time speech enhancement in the waveform domain,” 2020. [1](#), [2](#)
- [3] S. Pascual, A. Bonafonte, and J. Serrà, “Segan: Speech enhancement generative adversarial network,” 2017. [1](#)
- [4] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, “Speech enhancement based on deep denoising autoencoder,” 2013. [2](#)
- [5] S. Raj, P. Periasamy, and S. Gupta, “Audio signal quality enhancement using multi-layered convolutional neural network based auto encoder–decoder,” *International Journal of Speech Technology*, vol. 24, pp. 1–13, 06 2021. [2](#)
- [6] J. Chen, J. Benesty, Y. Huang, and S. Doclo, “New insights into the noise reduction wiener filter,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, pp. 1218 – 1234, 08 2006. [2](#)
- [7] J. Le Roux and E. Vincent, “Consistent wiener filtering for audio source separation,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 217–220, 2013. [2](#)
- [8] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, “Diffsound: Discrete diffusion model for text-to-sound generation,” 2022. [2](#)
- [9] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “Diffwave: A versatile diffusion model for audio synthesis,” 2020. [2](#), [3](#)
- [10] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” 2020. [2](#), [3](#)
- [11] S. T and G. TQ, “Toward a computational neuroethology of vocal communication: From bioacoustics to neurophysiology, emerging tools and future directions,” 2021. [3](#)
- [12] S. Han and J. Lee, “NU-wave 2: A general neural audio upsampling model for various sampling rates,” sep 2022. [3](#)