Nepali Text-to-Speech Synthesis using Tacotron2 for Melspectrogram Generation

Supriya Khadka[1*], *Ranju G.C.*[2*], *Prabin Paudel*[3*], *Rahul Shah*[4*], *Basanta Joshi*[5]

[12345]Department of Electronics and Computer Engineering
Institute of Engineering, Pulchowk Campus, Nepal

[1234]{075bct090.supriya, 075bct064.ranju, 075bct060.prabin,
075bct063.rahul}@pcampus.edu.np
[5]basanta@ioe.edu.np

Abstract

The paper proposes a method for generating high-quality synthesized Nepali speech from the text using the Tacotron2 model for melspectrogram generation. The speech synthesis process involves two phases: melspectrogram generation and vocoder output. The Nepali text is preprocessed and tokenized before being fed into a Tacotron2 model for generating melspectrograms. The Tacotron2 model is trained on a publicly available OpenSLR dataset for the Nepali language and finetuned on a new dataset created by the authors. Through fine-tuning, the model is refined to improve its performance and adapt it to language-specific characteristics. Further, incremental learning is employed to continually update the model with new data, ensuring its ability to generalize and adapt to evolving contexts. The melspectrograms are then sent to HiFiGAN and WaveGlow vocoders, which produce the synthesized speech. Finally, post-processing techniques are applied to further refine the generated output, enhancing its naturalness. The synthesized speech was qualitatively evaluated to obtain a Mean Opinion Score of 4.03 for naturalness, which stands as the highest among all previous Nepali Text to Speech tasks conducted to date.

Index Terms: Nepali Text-to-Speech Synthesis, Tacotron2, Melspectrogram, Vocoder

1. Introduction

Nepali, an under-resourced language in the field of text-to-speech (TTS) research, holds great potential for various applications. However, the existing TTS systems for Nepali fall short of delivering natural and fluent speech synthesis. Up until now, the approaches to generating speech from text have been traditional methods. This research aims to address the limitations of current systems using a modern TTS synthesis method and explore innovative approaches to create a more natural and intelligible Nepali TTS system, opening up new possibilities for improved accessibility, education, entertainment, and beyond.

In a traditional TTS system, the input text is first processed by a language analysis module that identifies the structure and pronunciation of the words and sentences. This processed text is then sent to a speech synthesis module that generates speech waveforms from the text. The speech synthesis module can use techniques such as concatenative, formant, and statistical parametric synthesis to generate the speech waveform.

Modern TTS systems use neural network-based models that can learn the relationship between the input text and the corresponding speech signal. These models are trained on large amounts of speech and text data and can produce high-quality synthesized speech that is indistinguishable from human speech in some cases. An encoder-decoder architecture is used in these systems. The encoder maps the input text into an intermediate representation, and the decoder generates the corresponding speech.

This TTS task can be broken down into two components:

- **Spectrogram Prediction Model**: The first component maps from strings of letters to mel spectrographs, which are sequences of mel spectral values over time. This is done using an encoder-decoder model, which takes the input text and generates the corresponding spectrogram as an output.
- **Vocoder**: The second component maps melspectrograms to waveforms. This process utilizes a vocoder that takes the spectrogram as input and generates the corresponding waveform as output.

The contributions of our paper are as follows:

- We trained Tacotron2 [1] to generate melspectrograms specifically for Nepali Text-to-Speech (TTS) and conducted experiments using both HiFiGAN and WaveGlow as vocoders. Through evaluation, we selected HiFiGAN as the preferred vocoder for our system.
- To address the scarcity of available datasets for Nepali TTS, we created a new dataset comprising approximately 1.2 hours of text and speech data. This dataset significantly contributes to the field as there is currently a lack of sufficient Nepali TTS datasets.
- In terms of performance evaluation, we achieved a high Mean Opinion Score (MOS) of 4.03. This MOS score stands as the highest among all previous Nepali TTS tasks conducted to date, demonstrating the effectiveness and quality of our proposed approach.

2. Background & related work

The development of natural-sounding Text-to-Speech (TTS) systems has made significant progress in widely spoken languages like English. However, creating such systems for under-resourced languages such as Nepali remains challenging. Previous endeavours in Nepali TTS include early efforts by a group of researchers from Columbia University, Carnegie Mellon University, and Cepstral LLC, who worked on developing phonetic lexicons [2]. Additionally, Bhasa Sanchar, a pioneering attempt from Nepal, utilized a concatenative approach with the Festival TTS system [3].

A concatenative speech synthesis approach produces sound by combining recorded sound clips. There are three main kinds of concatenative synthesis: Unit Selection Synthesis, Diphone Synthesis and Domain-specific Synthesis. Bhasa Sanchar was based on the unit selection synthesis method, which uses large databases of recorded speech and creates a database from the

*Equal Contribution

recorded utterance [4]. It was observed that long input to the engine caused unusual output which was messy, lacking proper intonation (too fast or too slow reading), and thus sounding unnatural. The qualitative evaluation was done by Mean Opinion Score(MOS), which was observed to be 3.2 [5].

An improvement to Bhasa Sanchar was developed in 2017 with modifications in the general architecture, by adding a tokenizer and a post-processing module [5]. A MOS of 3.6 was observed and it showed an overall improvement of 6 percent in terms of naturalness and intelligibility. Bhasa Sanchar also collaborated with the Sambaad project, which had initially been developing TTS for Nepali using the Festival and Festvox systems to create easy accessibility technological accessibility [6].

The Nepali TTS system trained on the Flite TTS engine gives the facility of changing pitch and speed, but it sounds robotic and is not completely noise-free [7]. Other Nepali TTS systems built with concatenative approaches employing Epoch Synchronous Non-Overlap Add Method (ESNOLA) and Time Domain Pitch Synchronous Overlap Add Method (TDPSOLA) are also in existence, with issues of their own. ESNOLA Nepali TTS system uses partneme for concatenation which makes the size of the speech dictionary quite small. There also might be some disturbance in the concatenation point of the synthetic speech signal [8]. TDPSCOLA Nepali TTS system has a small speech dictionary making it less accurate and error-prone [9].

A Nepali TTS system using a formant approach and FreeTTS synthesizer was developed in 2018 [10]. The formant approach is a rule-based synthesis method, which synthesizes speech using additive synthesis and an acoustic model taking multiple parameters [11]. However, these systems still need improvement in various speech synthesis aspects [12].

2.1. Tacotron2

Tacotron2[1] model is a deep neural network which uses a sequence-to-sequence architecture to generate speech from input text. It utilizes encoder-decoder architecture with an attention mechanism.

The encoder in Tacotron2 takes a sequence of letters as input and generates a hidden representation for the decoder's attention mechanism. It consists of character embeddings, convolutional layers to capture larger contexts, and a bidirectional LSTM (BiLSTM) that produces hidden states representing the character context.

The decoder in Tacotron2 generates melspectrogram output using the encoder's hidden representation. It consists of a dimensionality-reducing pre-net, a two-layer LSTM for hidden state generation, and a stop token prediction mechanism. The decoder LSTM takes input from the pre-net, previous attention context, and previously predicted melspectrogram. The stop token prediction is done by a linear layer with sigmoid activation.

After the decoding phase, the melspectrogram is fed into the Post-Net. It is a stack of convolution layers, which can capture the features from both past and future contexts in sequence.

In the context of Nepali Text-to-Speech (TTS), while WaveNet and Tacotron models have not been extensively utilized, there exists a work that explores Neural WaveNet Vocoder for Nepali TTS synthesis. This particular system preprocesses the Nepali Speech dataset obtained from OpenSLR and develops a Nepali TTS model using an Attention-Based Recurrent Sequence-to-Sequence architecture along with the WaveNet vocoder [13]. However, the quality evaluation of this system, measured through the Mean Opinion Score (MOS), yielded a score of 2.78, which is even lower than the MOS obtained from concatenative approaches [5]. This outcome can be attributed to various factors, including the limited size of the dataset, noise present in the dataset, and the lack of computational resources.

2.2. Vocoder

Vocoder analyzes and synthesizes human speech and other sounds. The primary function of a vocoder is to analyze an audio signal and separate it into two components: the spectral envelope, which contains information about the frequency content of the signal, and the excitation signal, which contains information about the sound source and the timing of the sound events. In our paper, Tacotron2 is accompanied by two vocoder models: HiFiGAN and WaveGlow.

HiFiGANs are Generative Adversarial Networks, built upon the principle of traditional GANs, which use a generator and a discriminator network to learn to generate realistic samples of speech signals [14]. WaveGlow is a flow-based network capable of generating high-quality speech from melspectrograms [15]. Through MOS evaluation, HiFiGAN was selected for our final system.

3. Experimental workflow

3.1. Data collection

The quality and quantity of data are fundamental to any deep learning system. In TTS, it's less crucial to use multiple voices, so basic TTS systems are speaker-dependent i.e. trained to have a consistent voice, on much fewer data, but all from one speaker [16]. We collected data from three different sources: *OpenSLR Dataset*, *Self-Recorded Data* and *Youtube Audiobook Clippings*.

3.1.1. OpenSLR dataset

We used *High Quality TTS data for Nepali* from OpenSLR[17] as our primary data source. It contains 2.8 hours of data having 2064 sentences and high-quality audio clips of 1-13 seconds associated with each sentence from 19 female speakers.

A few insights obtained from the data were:

1. The majority of data contains named entities like people's names, foreign transliterated words and countries and people's names
2. Many sentences are news articles like statements which might not favour conversation-like speech.
3. It contains many repetitive phrases.

3.1.2. Self recorded data

The data obtained from OpenSLR did not meet the requirements of prosody. So we recorded our audio clips in male and female voices. It contains 1.2 hours of data having 666 sentences and high-quality audio clips of 1-14 seconds from 2 speakers. It has 557 sentences of female recordings and 109 sentences of male recordings.

During the data recording process, we adhered to specific guidelines to ensure consistency and quality:

- We maintained a fixed distance between the recording device and the speaker, ensuring consistent audio capture conditions.
- Prior to and following each recording clip, a consistent duration of silence was maintained, allowing for adequate separation between consecutive utterances.
- To maintain audio uniformity, we carefully controlled the volume levels throughout each recorded clip.

- At the beginning of each clip, the speaker's voice was intentionally kept at a lower volume level to ensure a smooth and natural audio transition.

The data was obtained from three different sources

1. Publicly Available Nepali Essay Collection as the primary data source
2. Shwet Bhairabi e-book for enhancing Nepali vocabulary[18]
3. Nepali Newspaper Sentence Snippets for Out Of Vocabulary (OOV) words

Table 1: *OpenSLR vs. self recorded data*

	OpenSLR	**Self-Recorded**
Speakers	19	2
Audio Samples	2064	666
Gender	All Female	Male and Female
Total Duration	$\approx$ 2.8 hours	$\approx$ 1.2 hours
Total Words	17769	8529
Unique Words	6822	3982
Audio Clips Length	2-14 seconds	2-15 seconds

3.1.3. Youtube audio clippings

We briefly considered incorporating Audiobook Recordings sourced from YouTube. 160 audio clips were extracted through YouTube videos featuring book recitations. However, these clips were found to have prominent background music, which, despite undergoing audio processing, remained substantially present. This increased the noise from the model rendering the speech entirely unintelligible. Thus, we decided to exclude these clips from the official training dataset.

3.2. Data preprocessing

3.2.1. Audio preprocessing

1. The file type was converted to a common format of .wav from different types like .flac, .m4a, .mp3 and .mp4
2. The audio clips were converted from a sampling rate of 48kHz to 22.05kHz.
3. The stereo was set to mono.
4. Silence was removed from the beginning and end.
5. The long silences and other noises were removed from the middle of the audio clips.
6. Very long clips were clipped in between.

3.2.2. Text preprocessing

1. The numerals, years and dates are converted into text using a Python script.
2. Unwanted and unseen characters like symbols were removed
3. Lengthy sentences were broken down into smaller ones
4. Appropriate stop token was added manually at the end of each sentence

3.3. Training details

With a focus on the female voice, the training of the model is done in two phases. The first phase is pre-training with Nepali data, which is done with the high-quality TTS dataset from OpenSLR. The dataset was split into the train-to-validation ratio of 90:10 with a batch size of 4 and a learning rate of 0.001.

The result obtained from pre-training was understandable but lacked prosody. We adopted a data-driven approach for prosody management by recording audio with suitable intonations. The method of incremental learning was adopted for finetuning, where the self-recorded input data was continuously fed to extend the model's vocabulary and pronunciation dictionary. The dataset was split into a train-to-validation ratio of 90:10 with a batch size of 2 and a learning rate of 0.001. The result from HiFiGAN was superior to WaveGlow during the pre-training phase, hence only HiFiGAN was used for finetuning.

Table 2: *Audio parameters*

Parameter	**Value**
Max Waveform Value	32768
Sampling Rate	22.05 kHz
Filter Length	1024
Hop Length	256
Window Length	1024
No. of Mel Frequency Channels	80

The validation loss for pre-training was obtained to be 0.234 after 72 epochs. The best validation loss for finetuning was 0.328 after 68 epochs.

3.4. Audio postprocessing

The self-recorded data, not being captured in a studio environment, is not entirely noise-free. As a result, the finetuned audio is also noisy. So we applied a noise reduction procedure using a Python package. While this reduced the noise, it unintentionally affected the volume of different audio segments. To mitigate this, normalization was employed in the post-processing stage, ensuring a consistent volume level throughout all audio samples. Although the Python package provided satisfactory results for normalization, it had some limitations. Therefore, for future improvements and precision in volume control, we intend to develop a custom script to better manage the normalization process. The model also could not synthesize very long texts at once due to training data limitations. Hence the synthesized audio goes through a post-processing module, which comprises the following process:

1. Audio clips for longer texts are concatenated for the final result.
2. Noise is removed from the generated audio.
3. The noise-free audio is finally normalized.

We felt the need for post-processing only for our self-recorded data, hence it is not done for the initial training phase.

4. Result and analysis

4.1. Mean opinion score

Mean opinion score (MOS), currently considered the most reliable metric for evaluating speech synthesis [19], involves playing synthesized sentences to listeners and obtaining ratings on a scale of 1-5 to assess the quality of the synthesized utterances.

We conducted the MOS evaluation in two phases. In both phases, a certain number of audio clips were given in a Google Form format to a certain number of volunteers. They scored on the basis of Accuracy and Naturalness on a scale from 1-5, 1 being the worst and 5 being the best.

4.1.1. Result (MOS Phase 1)

The primary aim of this phase was to ascertain the required extent of model finetuning and to compare the performance of two vocoder models, namely WaveGlow and HiFiGAN. A total of 10 sample sentences were presented to 18 volunteers, with each vocoder used to generate the same sentence. The volunteers are school and college-level students and their parents, from the age group 13 to 60. The resulting scores were separately averaged to evaluate accuracy and naturalness.

Table 3: *MOS result 1: HiFiGAN vs WaveGlow*

	Naturalness	**Accuracy**
HiFiGAN	3.64	3.78
WaveGlow	3.47	3.68

HiFiGAN was seen to be the superior vocoder both in terms of accuracy and naturalness, hence it was used for finetuning.

4.1.2. Result (MOS Phase 2)

The second stage of MOS evaluation involved assessing the refined speech(first finetuned then post-processed) through a study with 28 participants. They were exposed to 20 audio clips and evaluated them based on accuracy and naturalness using a 1-5 scoring system. Finetuning yielded enhanced results in both naturalness and accuracy.

Table 4: *MOS result 2: HifiGAN*

	Naturalness	**Accuracy**
HiFiGAN	4.03	4.01

4.2. Melspectrogram analysis

A comparative analysis was conducted on the melspectrograms of synthesized voice before and after finetuning, along with a human voice on a Nepali phrase. The comparison considered various aspects such as Signal-to-Noise Ratio, Clarity and Dynamic Range. The human speech exhibited a wide frequency range from 0-20 kHz, while the synthesized speech had a narrower range from 0-11 kHz. Even in the lower frequencies where human ears are more sensitive, human speech had a lower signal-to-noise ratio, followed by finetuned speech.

The audio of finetuned model closely resembled human speech as compared to the audio before finetuning as indicated even by melspectrogram. The distinct separation between words resulted in a higher quality of speech, whereas melspectrogram of initial audio has smeared spectrogram which causes certain features to overlap. Additionally, in high-quality speech, each element had a high intensity at a specific frequency and clear separation between frequencies. Enhancements in naturalness and differentiation were achieved through finetuning, but further improvements are necessary to extend the frequency range and enhance overall quality.

4.3. Accuracy estimation

The mean opinion score (MOS), being a subjective measure, has limitations in accurately determining speech accuracy. To address it, we adopted a manual checking mechanism to estimate accuracy. This method is corresponding to Word Error Rate (WER) for Automatic Speech Recognition(ASR). In this process, 80 audio clips, containing sentences with an average length of 17 words, were listened to. The average error rate was found to be 2.4 per sentence. The figure above highlights the words that our model struggled to pronounce correctly. Using this approach, we achieved an accuracy of 86%. However, it should be noted that this method still relies on subjective judgment and there is considerable room for further improvement.

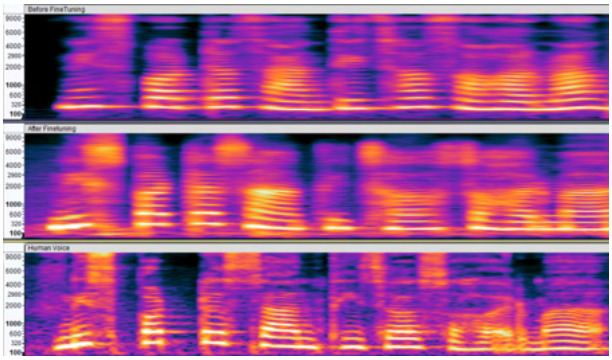

Figure 1: *Melspectogram comparison*

राधा कस्ती **सरल** बालिका जस्ती थिई, उसको गोल-गोल अनुहारमा अझै बालिकाको **स्निग्धता** बाँकी नै थियो- तर आमा पनि हुन लागेकी थिई **अब त्यो** ।
अझै गाउहरुमा छोराले विद्यालयमा पढ्न पाउने तर छोरीले घरमा काम गर्नु पर्ने हुन्छ ।
आमाले जे जस्तो **खान** दिनुहुन्थ्यो म त्यही खाइ दिन्थे, तर **दुधदही** भनेपछि चाहिँ म अलि लोभिन्थे ।
वनबाट नै पशुहरूलाई घाँस **प्राप्त** हुन्छ ।

Figure 2: *Accuracy estimation sample*

5. Conclusion

In conclusion, this research paper presents an approach for generating high-quality synthesized Nepali speech using the Tacotron2. Through a two-phase process involving melspectrogram generation and vocoder output, the text-to-speech synthesis system effectively converts Nepali text into natural-sounding speech. The Tacotron2 model is trained on both a publicly available OpenSLR dataset for the Nepali language and a new dataset specifically created for this study. The melspectrograms generated by the model are then fed into a HiFiGAN vocoder, which produces the final synthesized speech. The qualitative evaluation of the synthesized speech resulted in an impressive Mean Opinion Score of 4.03 for naturalness and 4.01 for accuracy, surpassing the performance of all previous Nepali Text-to-Speech systems developed to date. The accuracy estimation also showed a promising result of 86%. This achievement highlights the effectiveness of the proposed method and its potential for various Nepali Text-to-Speech applications.

6. Acknowledgements

We would like to express our sincere gratitude towards the faculty members of the Department of Electronics and Computer Engineering, IOE Pulchowk Campus for their constant support and guidance and the American Society of Nepalese Engineers for providing a grant to conduct this research.

7. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[2] S. R. Maskey, A. W. Black, and L. M. Tomokiyo, "Bootstrapping phonetic lexicons for new languages," 2004.

[3] "Language technology kendra projects," [Date Accessed: May, 2023]. [Online]. Available: https://ltk.org.np/projects.php

[4] R. Bajracharya, S. Regmi, B. K. Bal, and B. Prasain, "Building a natural sounding text-to-speech system for the nepali language: research and development challenges and solutions," *Gipan*, vol. 4, pp. 106–116, 12 2019.

[5] R. R. Ghimire and B. K. Bal, "Enhancing the quality of nepali text-to-speech systems," in *Creativity in Intelligent Technologies and Data Science*, A. Kravets, M. Shcherbakov, M. Kultsova, and P. Groumpos, Eds. Cham: Springer International Publishing, 2017, pp. 187–197.

[6] S. Dhakhwa, P. A. Hall, G. B. Ghimire, P. Manandhar, and I. Thapa, "Sambad-computer interfaces for non-literates," *Lecture Notes in Computer Science*, vol. 4550, p. 721, 2007.

[7] K. Subedi, "Nepalispeech - convert written nepali text to speech." [Online]. Available: https://nepalispeech.com/

[8] B. Chettri and K. Shah, "Nepali text to speech synthesis system using esnola method of concatenation," *International Journal of Computer Applications*, vol. 62, pp. 24–28, 01 2013.

[9] P. Malla, "Nepali text to speech using time domain pitch synchronous overlap add method," Master's thesis, 2015.

[10] K. B. Shah, K. K. Chaudhary, and A. Ghimire, "Nepali text to speech synthesis system using freetts," *SCITECH Nepal*, vol. 13, no. 1, pp. 24–31, 2018.

[11] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," *the Journal of the Acoustical Society of America*, vol. 67, no. 3, pp. 971–995, 1980.

[12] T. B. Shahi and C. Sitaula, "Natural language processing for nepali text: a review," *Artificial Intelligence Review*, pp. 1–29, 2022.

[13] A. Banset, B. Joshi, and S. Sharma, "Deep learning based voice conversion network," 10 2021, p. 1292 – 1298.

[14] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[15] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[16] D. Jurafsky and J. H. Martin, *TTS*. Pearson, 2022.

[17] K. Sodimana, K. Pipatsrisawat, L. Ha, M. Jansche, O. Kjartansson, P. D. Silva, and S. Sarin, "A Step-by-Step Process for Building TTS Voices Using Open Source Data and Framework for Bangla, Javanese, Khmer, Nepali, Sinhala, and Sundanese," in *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, Gurugram, India, Aug. 2018, pp. 66–70. [Online]. Available: http://dx.doi.org/10.21437/SLTU.2018-14

[18] "Shwet bhairabi: Free download, borrow, and streaming." [Online]. Available: https://archive.org/details/shwet-bhairabi_202204

[19] D. Jurafsky and J. H. Martin, *TTS Evaluation*. Pearson, 2022.

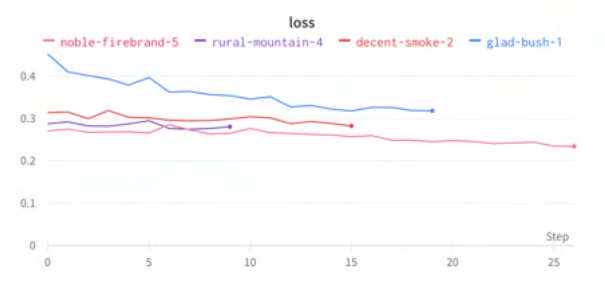

Figure 3: *Validation loss for pretraining*

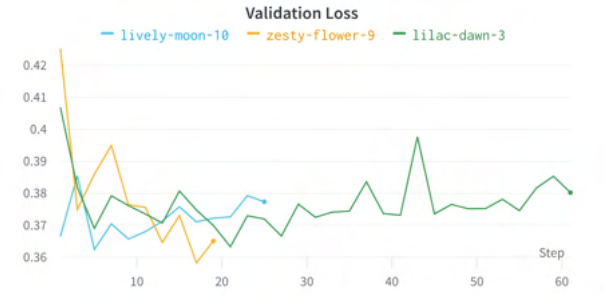

Figure 4: *Validation loss for finetuning*

A. Appendix

A.1. Training results

Each line graph in the above figure shows different training periods of incremental learning. The validation loss fluctuations during the finetuning process can be attributed to the non-studio environment in which the recordings were made. The presence of noise in the input signal resulted in an output that also contained noise, leading to the observed variability in the validation loss.

A.2. Audio output

The audio output samples can be found at `https://shruti-audios.netlify.app/`