Development of a Language Text-to-Speech System Indonesia with Characteristic Control based on Prompt

M. Alif Al Hakim1, Winoto Hasyim2, Matthew Hotmaraja Johan Turnip3
1Faculty of Computer Science, University of Indonesia, Depok, 16424, email: malif.al@ui.ac.id
2Faculty of Computer Science, University of Indonesia, Depok, 16424, email: winoto.hasyim@ui.ac.id
3Faculty of Computer Science, University of Indonesia, Depok, 16424, email: matthew.hotmaraja@ui.ac.id

ABSTRACT — Prompt-based Text-to-Speech (TTS) systems offer the ability to generate audio with voice characteristics. which can be customized according to text descriptions. This technology has the potential to support a variety of applications, such as language learning and accessibility of information. However, the development of TTS models for Indonesian faces challenges, especially in producing high-quality audio that matches the text description. This problem is exacerbated by the limited Indonesian language dataset with variations adequate voice. This study proposes Parler-TTS, a prompt-based model consisting of a text encoder to capture information semantics, an autoregressive decoder with cross-attention mechanism to generate audio tokens, and a DAC as an audio codec for sound wave reconstruction. The model was trained on an Indonesian audio dataset collected via Youtube. Before use To train the model, the dataset is preprocessed and annotated automatically first using several approaches to measure audio quality and characteristics. The model is trained using parameter initializations from the English-language model to speed up convergence. Hyperparameter adjustments were also performed to improve performance. Experimental results show that the CLAP score similarity (0.21) and Word Error Rate (51%) are still relatively low. However, subjectively the model produces audio with a score MOS is quite good for clarity (3.825) and relevance to the prompt (4.375). This study provides contributions in the form of development of a customizable prompt-based TTS model, as well as a comprehensive Indonesian audio dataset, opens up opportunities for further development.

KEYWORDS — Text-to-Speech, Prompt, Characteristics, Voice, Cross-Attention

I. INTRODUCTION

Text-to-Speech (TTS) technology has experienced rapid development in recent years with a variety of applications spanning education, entertainment, and assistive technology. In this study, we explore development of TTS system based on voice characteristic control use text descriptions or prompts to generate audio that suits the user's needs. With this approach, users can specify attributes such as speech rate, intonation, and pitch directly through the text. descriptive.

Although various TTS models have reached a level of good naturalness and clarity, ability to control specific sound characteristics through prompts text is still limited. The main problem lies in the limited Indonesian language dataset with various voices. inadequate, which often results in less than optimal model performance in generating audio with the desired attributes accurate as per description.

This study proposes the Parler-TTS model, a TTS system prompt-based designed to produce audio according to the user description. This model consists of three main components: text encoder, decoder, and audio codec. Text The encoder uses the Flan-T5 architecture to convert text description into semantic representation. Representation is used to initiate the generation of audio tokens. on an autoregressive decoder equipped with a mechanism cross-attention. This mechanism ensures that information is The text input is optimally utilized during the voice generation process, so that the resulting audio characteristics are consistent with the user's description. The last component is Descript Audio Codec (DAC) based audio codec, which serves to reconstruct sound waves from audio tokens.

The training process is carried out using a dataset annotated and processed Indonesian language audio extensively. The processing steps include text normalization, voice feature extraction, and description generation. based on the sound characteristic category. To accelerate convergence, the model is initialized with predetermined parameters. trained using English dataset before proceeding with fine-tuning on Indonesian language data. Hyper-parameter tuning was also applied to optimize stability and efficiency of training, with adjustments to learning rate, batch size, and number of epochs.

Evaluation is done using metrics such as Mean Opinion Score (MOS) for audio quality, CLAP similarity for audio conformity to description, and Word Error Rate (WER) for word pronunciation accuracy. Experimental results show that the model is capable of producing audio with MOS values of 3,825 for clarity and 4,375 for conformity with the text description. However, the performance on objective metrics such as CLAP similarity (0.21) and WER (51%) still need improvement. These results show potential for the model to be further improved with datasets more diverse, high quality, and training that longer.

This research provides a major contribution to the development of a prompt-based TTS system that allows control of voice characteristics in Indonesian. The proposed methodology includes the integration of innovative model architecture with comprehensive dataset processing, and performance evaluation using subjective and objective metrics.

This research opens up new opportunities for the development of TTS systems that are more adaptive and suited to local needs.

This report is organized as follows: Section I discusses the introduction and objectives of the study. Section II summarizes the related literature review. Section III describes the proposed methodology, including the model architecture and training process. Section IV presents the experiments and analysis of the results.
Section V concludes the main findings and provides recommendations for future research.

II. LITERATURE REVIEW

The development of TTS systems with prompt-based control capabilities has become an increasingly important research area to meet the needs of flexibility and high audio quality. Research by Lyth and King [1] introduces a prompt-based approach to control voice characteristics, such as speaker identity, speaking style, and acoustic conditions. Their model leverages synthetic annotations on a large-scale dataset (45 thousand hours) to train a speech language model. This approach allows intuitive control through natural language descriptions, producing high-quality speech without the need for large-scale manual labeling.

Unlike traditional methods such as global style tokens [2] or reference embeddings [3] that rely on reference recordings to control voice style, the prompt-based approach used by Lyth and King provides greater flexibility. By replacing manual annotation with automated annotation, they are able to use large-scale datasets without the constraints of human labeling. Other prompt-based systems, such as Audiobox [4] and PromptTTS2 [5], have also explored the use of natural descriptors, but their control capabilities are limited to specific attributes such as emotion or speech rate. The smaller datasets of these systems also limit the scale and flexibility compared to the model proposed by Lyth and King.

Additionally, other research shows great potential in prompt-based generation, as demonstrated by Mu-sicGen [6]. MusicGen uses a combination of textual descriptions and melodic features to generate customizable music, with innovations such as MusicGen-Chord and MusicGen-Remixer. This modular approach shows how prompt-based control can be applied to generative tasks with a high degree of precision, an idea that is relevant to TTS.

However, most of the prompt-based TTS research is still limited to English. For example, LibriTTS-P [7] provides an annotated dataset for English with description of speaking style and speaker characteristics. This dataset allows the model to generate more natural and relevant speech with text descriptions. Until now, a similar approach has not been extended to Indonesian, which presents unique challenges such as complex morphological structures, distinctive phonology, and different intonation patterns.

This study aims to adapt the prompt-based model to Indonesian by addressing the existing linguistic challenges. The developed system utilizes an encoder-decoder architecture with a cross-attention mechanism to generate speech that matches the text description. The main innovation of this study includes the integration of synthetic annotations to create a local dataset that reflects the cultural and linguistic needs of Indonesian. By adapting a modular approach such as that used in MusicGen and the prompt-based method of Lyth and King, this study seeks to provide a more relevant solution to local needs and expand the scope of applications of prompt-based TTS.

III. METHODOLOGY

A. Parler TTS Model

The Parler-TTS model [1] has an architecture similar to the MusicGen model architecture [8]. This model has three main parts, including the following:

1) **Text encoder:** This part functions to convert the prompt description of the sound characteristics into a hidden-state representation. Parler-TTS uses a frozen text encoder initialized from the Flan-T5 model [9].

2) **Parler-TTS decoder:** This part is a language model (LM) that autoregressively generates audio tokens (or audio codes) based on the hidden state representations generated by the text encoder. The decoder is designed to use a cross-attention mechanism to use information from the text representations during the audio generation process.

3) **Audio codec:** This part is used to recover the audio waveform from the audio tokens predicted by the decoder. Parler-TTS uses the DAC (Descript Audio Codec) model [10] for this part.

Although the architecture of Parler-TTS is similar to MusicGen, there are some modifications introduced, including the use of cross-attention to enrich the audio output with text description information. The processed text description will be combined with the decoder input hidden states or sentences to be converted into audio. In addition, Parler-TTS uses DAC as an audio encoder because it provides better sound quality compared to EnCodec [11].

B. **How the Parler TTS model works** The

Parler-TTS model is designed to generate audio with voice characteristics that can be controlled using prompt-based text descriptions. How this model works starts

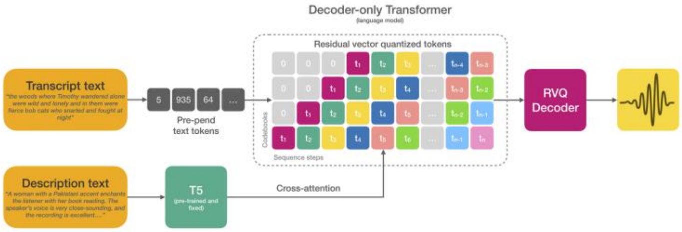
Decoder-only Transformer
(language model)
Residual vector quantized tokens
Transcript text
"the woods where Timothy wandered alone were wild and lonely and in them were fierce bob cats who snarled and fought at night"
5
935
64
...
Pre-pend text tokens
Codebooks
Sequence steps
Description text
"A woman with a Pakistani accent enchants the listener with her book reading. The speaker's voice is very close-sounding, and the recording is excellent...."
T5
(pre-trained and fixed)
Cross-attention
RVQ Decoder

Figure 1: Overview of the Parler-TTS model architecture

by processing input in the form of detailed text descriptions, such as "In a calm and slow voice, with a slight echo in the background". This description is given to the Text Encoder component, which is tasked with converting the text into a hidden-state representation. Parler-TTS uses Flan-T5, a pre-trained Transformer-based en-coder, which is able to capture semantic information from text descriptions efficiently. The resulting hidden-state representation then becomes the initial condition for the audio token generation process.

In the next stage, this hidden-state representation is fed into the Decoder, an auto-regressive language model that incrementally generates audio tokens.
The Parler-TTS decoder uses a cross-attention mechanism, which allows the model to integrate information from text descriptions during the audio generation process. With this mechanism, the model is able to ensure that the characteristics of the generated voice, such as speech rate, intonation, reverberation, and pitch, match the description provided by the user. This process is done in an autoregressive manner, where each audio token is predicted based on previous tokens until the entire sequence of audio tokens is complete.

The final stage is the reconstruction of the sound wave from the generated audio tokens. Parler-TTS uses DAC, an audio codec model designed to produce high-quality sound waves with sampling rates up to 44 kHz. The DAC takes the audio tokens from the decoder and converts them into a final audible audio file.
With this architecture and workflow, Parler-TTS is able to produce audio that is not only realistic but also matches the characteristics desired by the user.

In the training process, features derived from prompts and hidden-state representations and audio tokens generated from the owned groundtruth audio will be used. After that, the training process will be carried out on the decoder model which also uses cross-attention to find the relationship between audio tokens and hidden-state representations. The model calculates an attention score for each pair of audio tokens and hidden-state representations. This attention score shows how relevant each audio token is to each part of the hidden-state representation. The attention score is used to weight the hidden-state representation related to an audio token. Higher weights are given to more relevant parts. Training uses a cross-entropy loss function which aims to minimize errors between the predicted audio token and the target audio token.

IV. EXPERIMENT

A. Dataset Preparation and Processing

The dataset used to train and test the model is a collection of Indonesian-language audio obtained from YouTube videos belonging to Eno Bening and Justinus Lhaksana.

1) **Obtaining Transcripts**

Transcripts are automatically obtained using a script that leverages an API-based speech-to-text service. The output of this process is two types of files: a full audio file containing the entire audio recording, and an SRT file containing timestamps and text transcripts corresponding to specific parts of the audio. These SRT files identify specific time ranges in the audio recording, with each audio segment annotated with the corresponding transcript for that timestamp range.

2) **Audio Data Organization**

Each separate audio part based on the timestamp in the SRT file will be extracted into a separate audio file with its own unique ID or file name.
After that, several parts or audio clips will be combined so that they last about 8 - 12 seconds. The file name is used to identify and store each separate audio segment. This process aims to make it easier to manage and process smaller audio files. Next, to organize the audio data and its transcripts, a CSV file will be created. Each row in the CSV file will contain the name of the audio file and the transcript that corresponds to the audio segment of the file.
This process allows for grouping and processing smaller, more focused audio segments, allowing the model to more easily learn the characteristics of sounds in a specific time context. The total number of audio clips collected was 25,550 or equivalent to 65 hours.

3) **Transcript Text Normalization**

Text normalization aims to transform raw text into a more uniform format and ready to be processed by the TTS model. This process includes several steps, including number expansion, lower casting, removal of useless or redundant punctuation, and removal of excess spaces. This is necessary to ensure that the text input given to the TTS model is in an optimal form and can be processed consistently, thereby improving the accuracy and quality of the resulting voice.

4) **Extract Voice Characteristic Features**

To produce voice characteristics that can be customized through prompts, each audio needs to have its voice characteristics extracted first. The voice characteristics that are extracted include speaking rate, recording quality, speech quality, and utterance pitch. a)

Speaking Rate

Refers to the speed of speech in audio,

which affects how fast or slow speech is delivered. Extracting the speaking rate is important to allow for tempo adjustments in the speech output. It is calculated by divide the number of phonemes by the length of the utterance. In this study, the library used Indonesian Grapheme-to-Phoneme (G2P-id) [12].

b) **Recording Quality**

The quality of audio recordings can be measured objectively using the SNR (Signal-to-Noise Ratio) and C50 parameters. SNR indicates the ratio between the strength of the sound signal and the background noise, while the C50 measures reverberation or echo time in a room. Brouhaha-like voice activity detection model from pyannote [13] can be used to calculate these two parameters. High SNR values indicates minimal noise, while the value A low C50 indicates little echo. This information is very useful for evaluating recording quality.

c) **Speech Quality**

To evaluate the quality of a speech signal objectively, various methods can be used. metrics provided by Torchaudio. Torchau-dio Speech Quality and Intelligibility Measures [14] offers a series of tools to calculate important metrics such as PESQ, STOI, and SD-SDR.

i) **Perceptual Estimation of Speech Quality (PESQ):** The PESQ metric is designed to mimic human perception of speech quality. PESQ provides a numeric score between -0.5 up to 4.5, where the score is higher shows better quality. PESQ very sensitive to distortions that affect the intelligibility and naturalness of speech

ii) **Short-Time Objective Intelligibility (STOI):** STOI focuses on the intelligibility aspect of speech, namely how easily an utterance can be understood. STOI calculates the correlation between degraded speech spectrogram with clean speech spectrogram. The STOI value ranges from between 0 to 1, where the value is close 1 indicates high intelligence.

iii) **Scale-Invariant Signal-to-Distortion Ratio (SD-SDR):** SD-SDR is a more advanced metric new and considered more robust than traditional metrics such as SDR. SD-SDR measures how much of the original signal is still remaining in the degraded signal, taking into account its scale. The SD-SDR value higher indicates less distortion.

d) **Pitch Utterance**

In this study, pitch utterance was analyzed or the average pitch of each audio clip. Pitch is an acoustic characteristic that determines the pitch of a sound. By measuring pitch utterance, information can be obtained about the speaker's intonation. To calculate the pitch utterance on each audio clip, using the model Pitch-Estimating Neural Networks (PENN) [15]. After obtaining the pitch value for each point time in the audio clip, two statistics will be calculated main, namely the average and standard deviation of the pitch utterance.

5) **Changing Continuous Representation of Characteristics Voice Becomes Categorical**

The previously extracted continuous sound characteristics are transformed into categorical representations using a discretization method. This process involves dividing the continuous value range into seven intervals. called bins. Histogram of the distribution of characteristic values used to determine the bin boundaries optimally. Each continuous value is then mapped to a bin. appropriate, and given the appropriate category label predetermined. For example, for the speaking rate characteristic, the seven possible categories are very slow, a bit slow, a little slow, speed medium, slightly fast, somewhat fast, and very fast. With this categorical representation, users can easily easy to choose the desired speaking rate, thus facilitating the control and modification process sound output.

6) **Create a Description for Each Audio based on Sound Characteristics Category**

Each audio will be given a description. automatically generated based on category characteristics of its sound. To achieve this, Large Language Model (LLM) Gemma-2B [16] is used. The categories of voice characteristics will be given as input (prompt) to the model, and The model will produce descriptive text as output. For example, if the input is “expressive voice, very slow speaking, slightly noisy”, then the output is probably is” In a very expressive voice, Jenny pronounces her words incredibly slowly. There's some noise in the recording”

B. **Training Process**

The training process is carried out on a notebook that has access to CUDA-based GPUs to speed up the process training. To train the model, it is first carried out dataset division into train and test with a ratio of 8:2. Before starting training, weight initiation will be carried out on Parler-TTS model pre-trained on a language dataset English. This is necessary so that the model does not have to learn from early and can utilize the knowledge that has been gained previously in studying the new dataset. After that, training or fine-tuning will be carried out on the Parler-TTS model.

Hyperparameter tuning is one way to improve model performance. Learning rate is a hyperparameter that regulates how many steps the model takes to update its weights during the training process. A learning rate that is too large can cause the model to be unstable and difficult to reach the optimal point, while a learning rate that is too small will slow down the training process. Choosing the right learning rate aims to ensure that the model learns effectively and stably.

The number of epochs determines how many times the entire dataset is used to train the model. The optimal number of epochs can help the model learn the data well and efficiently to avoid underfitting or overfitting.

Batch size is the number of data samples processed in one iteration before the model updates its weights. The batch size setting affects how the model learns the details and variations of the data, but the batch size also affects the number of iterations required to complete one epoch.

Gradient accumulation steps is a method to accumulate gradients from multiple iterations before performing weight updates. This technique allows for larger batch size simulations without requiring additional memory, making it particularly useful in resource-constrained environments. Properly setting gradient accumulation steps can improve training efficiency.

C. **Evaluation**

Evaluation will be conducted using several metrics, including Mean Opinion Score (MOS), CLAP similarity, and WER metrics. MOS is a crowdsourced metric that assesses the synthesis results based on clarity, naturalness, understandability, and expression. The MOS evaluation method is a subjective evaluation method where listeners are asked to score the quality of the audio produced by the TTS model. CLAP similarity [17] is a metric used to measure the suitability between text descriptions and the resulting audio. This metric calculates a similarity score between audio and text, where a higher score indicates that the resulting audio is more in line with the text description. In this study, the CLAP model was used on huggingface "laion/larger-clap-music-and-speech". Meanwhile, WER is a standard metric in speech recognition, but it is also relevant for evaluating text-to-speech in the context of understanding and reproducing words. This metric measures word recognition errors by comparing the transcripts produced by the model to the reference text. In this study, the Whisper model [18] was used which has been fine-tuned in Indonesian. The model used is the huggingface model "cahya/whisper-medium-id".

V. RESULTS AND ANALYSIS

The model that has been trained and optimized using the training method explained previously is then... conduct an evaluation. For an objective evaluation itself, the model gets a CLAP similarity value of 0.21. This result is still not very good because the similarity is low and also tends to be lower than the experiment conducted in paper [1]. For the WER metric, the model gets a value of 51 percent. This result is also still not optimal. However, when viewed qualitatively and subjectively, the Mean Opinion Score (MOS) value for audio clarity (3,825) and similarity to the description (4,375) shows that the model is able to produce audio that is quite understandable and relevant to the description input.

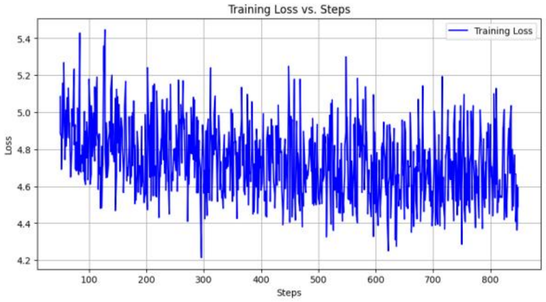

Figure 2: Model training graph

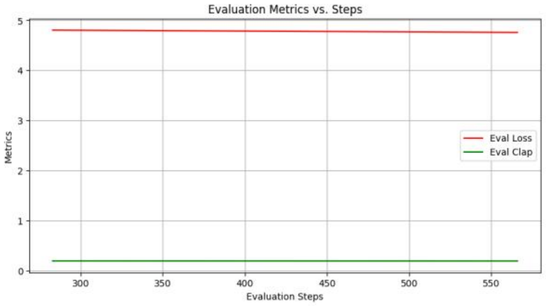

Figure 3: Model evaluation graph

In Figures 2 and 3, it can be seen that the model is still in the learning process. This can be seen from the training loss and validation loss trends that consistently decrease, although little by little. In addition, the clap similarity evaluation results also tend to increase, which indicates that the resulting audio is increasingly similar to the description, although not significantly. Therefore, there is an indication that the training process can still be continued for longer. With the availability of more adequate computing resources, the potential for improving model performance is still open.

When viewed based on WER and then analyzed qualitatively, there are still some audios whose pronunciation is not correct and not clear. This can be caused by several causes, including:

1) There are some inaccurate transcripts, such as some typos or mistyped words. This causes the model to have difficulty pronouncing some words.

2) Lack of sound variation in the dataset. This causes the model to have difficulty in generating audio that has characteristics that rarely appear.

In Figure 4, it can be seen that the expressive voice characteristics are very few when compared to other voice characteristics. This makes it difficult for the model to produce sounds with such voice characteristics clearly.

3) The transcripts in the dataset do not have punctuation such as commas, question marks, or periods. This causes the model not to learn when a sentence should pause or use the appropriate tone.

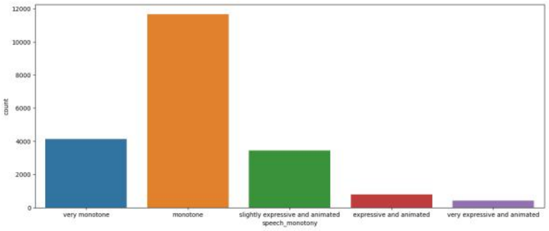

Figure 4: Distribution of sound characteristics

VI. CONCLUSION

For now, the performance of the parler-TTS model using the Indonesian language dataset is still relatively poor. The model still needs to adapt to Indonesian language data first. Using a quality dataset can help improve the quality of the resulting audio.

In addition, adding more data or adding training iterations can also be an approach to creating a better model that is familiar with the Indonesian language.

REFERENCE

[1] D. Lyth and S. King, “Natural language guidance of high-fidelity text-to-speech with synthetic annotations,” 2024. [Online]. Available: https://arxiv.org/abs/ 2402.01912 [2] Y. Wang, D.

Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R.A. Saurous, “Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis,” 2018. [Online]. Available: https://arxiv.org/abs/1803.09017 [3]

R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, RJ Weiss, R. Clark, and R.A. Saurous, “Towards end-to-end prosody transfer for expressive speech synthesis with tacotron,” 2018. [Online]. Available: https://arxiv.org/abs/1803.09047

[4] A. Vyas, B. Shi, M. Le, A. Tjandra, Y.-C. Wu, B. Guo, J. Zhang, Hsu, “Audiobox: Unified audio generation with natural language prompts,”

2023. [Online]. Available: https://arxiv.org/abs/2312.15821 [5]

Y. Leng, Z. Guo, K. Shen, X. Tan, Z. Ju, Y. Liu, Y. Liu, D. Yang, L. Zhang, K. Song, L. He, X.-Y. Li, S. Zhao, T. Qin, and J. Bian, “Promptts 2: Describing and generating voices with text prompts,” 2023. [Online]. Available: https://arxiv.org/abs/2309.02285 [6]

J. Jung, A. Jansson, and D. Jeong, “Musicgen-chord: Advancing music generation through chord progressions and interactive web-ui,” 2024. [On line]. Available: https://arxiv.org/abs/2412.00325 [7]

M. Kawamura, R. Yamamoto, Y. Shirahata, T. Hasumi, and K. Tachibana, “Libritts-p: A corpus with speaking style and speaker identity prompts for text-to-speech and style captioning,” 2024. [On line]. Available: https://arxiv.org/abs/2406.07969

[8] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Defossez, “Simple and controllable music generation,” 2024. [On line]. Available: https://arxiv.org/abs/2306.05284 [9]

HW Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, SS Gu, Z. Dai, M. Suzgun, language models,” 2022. [Online]. Available: https:// arxiv.org/abs/2210.11416 [10] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved rvqgan,” 2023. [Online].

Available: https://arxiv.org/abs/2306.06546 [11] A.

Defossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” 2022. [Online]. Available: https://arxiv.org/abs/ 2210.13438

[12] W. Wongso, A. Joyoadikusumo, and S. Limcorn, “g2p id,” https://github. com/ bookbot-kids/g2p id, 2024.

[13] M. Lavechin, M. Metais, H. Titeux, A. Boissonnet, J. Copet, M. Rivi ere, E. Bergelson, A. Cristia, E. Dupoux, and H. Bredin, “Brouhaha: multi-task training for voice activity detection, speech-to-noise ratio, and c50 room acoustics estimation,” 2023. [Online]. Available: https://arxiv.org/abs/2210.13248 [14] A. Kumar, K. Tan, Z. Ni, P. Manocha, X. Zhang, E. Henderson, and B. Xu, “Torchaudio-squim: Reference-less speech quality and intelligibility measures in

torchaudio,” 2023. [Online]. Available: https://arxiv.org/abs/2304.01448 [15] M. Morrison, C. Hsieh, N. Pruyne, and B. Pardo, “Cross-domain neural pitch and periodicity estimation,” 2024. [Online]. Available: https:// arxiv.org/abs/2301.12258 [16] G. Team, “Gemma,” 2024. [Online]. Available: https://www.kaggle.

com/m/3301

[17] B. Elizalde, S. Deshmukh, M.A. Ismail, and H. Wang, “Clap: Learning audio concepts from natural language supervision,” 2022. [Online]. Available: https://arxiv.org/abs/2206.04769

[18] A. Radford, JW Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” 2022. [Online]. Available: https://arxiv.org/abs/2212.04356