Universiteit Leiden

Master Computer Science

Unlocking Human-Like Speech: Enhancing TTS with Predicted Prosodic labels from text

Name:	Shubham Bhatt
Student ID:	s3287467
Date:	[29/09/2023]
Specialisation:	Master's in Computer Science: Data Science
1st supervisor:	Dr.ir. D.J. Broekens
2nd supervisor:	Dr. E.M. Bakker

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Acknowledgements

I would like to express my heartfelt gratitude to all the individuals who have contributed to the completion of this thesis. First and foremost, I extend my sincere appreciation to my supervisor, Joost for his support, guidance, and valuable insights throughout this research journey. His expertise and encouragement have been instrumental in shaping this thesis and enhancing my understanding of the subject matter.

I am indebted to my friends and colleagues for their encouragement during these challenging times. Their support and constructive feedback have been invaluable in refining my ideas and improving the quality of this thesis. I extend my heartfelt thanks to my parents and brother for their unwavering love, understanding, and encouragement. Their belief in me and constant motivation have been the driving force behind my academic pursuits. The completion of this thesis would not have been possible without the collective support and contributions of all those mentioned above. I am truly grateful for their invaluable assistance and encouragement, and I humbly dedicate this work to them.

Abstract

Text-to-speech (TTS) technology, despite its advancements, often produces speech that lacks the naturalness inherent in human communication, primarily due to insufficient incorporation of prosodic elements such as intonation and stress. This research aimed to enhance the naturalness of synthesized speech by integrating prominence labels into the Tacotron2 architecture, specifically targeting improved prosody in the generated speech. Various methods were devised to integrate these labels, including: incorporating them into input tokens for the encoder layer, concatenating them in the Attention layer, and predicting them analogously to the architecture's stop token loss. These methods were embodied in experimental setups, namely ENCO-MOD, ATT-MOD, and PRO-MOD, and were evaluated using objective metrics to assess their efficacy. Our experiments encountered significant challenges, particularly alignment issues while using a multi-speaker dataset, which yielded inconclusive results. A shift to a single-speaker dataset illuminated potential, with ATT-MOD showing notable alignment during training. However, the PRO-MOD, while able to produce synthesized speech with some degree of emphasis, demonstrated that extended training periods are crucial to fully ascertain its capability and reliability in generating prominently emphasized speech. These findings, although preliminary and constrained by the specified experimental setups, illuminate avenues for further research and model refinement, especially concerning extended training for PRO-MOD.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In an era where artificial intelligence strives to imitate human abilities, one area stands out as a fascinating frontier - Text-to-Speech (TTS) technology. It's the science and art of transforming inert, written text into dynamic, lifelike speech. But while TTS can read aloud a book or guide you via GPS, reproducing human-like sounding speech remains a challenge [24].
Recent advancements in artificial intelligence and machine learning have significantly propelled the field of Text-to-Speech (TTS) technology [3]. These developments have enabled modern TTS systems to generate synthesized speech that matches the naturalness, expressiveness, and variability of human speech.
Current TTS engines have advanced capabilities, including the ability to reproduce various speaking styles, diverse accents, and a range of emotional tones. This results in a more engaging and personalized user experience.
The use of TTS technology spans multiple domains, including audiobooks, navigation systems, voice assistants, and assistive tools for individuals with visual impairments. As the technology advances, it is anticipated to unlock novel applications, thereby transforming our interaction with technology and modes of information consumption [23]
Despite considerable advancements in text-to-speech (TTS) technology, there remains a notable gap in achieving truly natural-sounding synthesized speech. The concept of naturalness in TTS refers to the ability of the generated speech to exhibit prosodic features that are characteristic of human speech, including intonation, stress, rhythm, and variations in speed [20]. While TTS systems have made significant progress in producing intelligible and coherent speech, the challenge lies in capturing the subtleties and nuances that make speech sound natural. Prosodic features play a crucial role in conveying meaning, emotions, and emphasis in spoken language. However, current TTS systems often struggle to replicate these features accurately, resulting in speech that may sound robotic or monotonous.
The drawbacks associated with the current TTS technology manifest prominently in the form of decreased user engagement and comprehension, which consequently results in lower user satisfaction. The synthesized speech, often characterized by a robotic tone, tends to engender feelings of artificiality and detachment, thus posing a challenge for listeners to engage fully with the synthesized content [9]. Moreover, the absence of naturalness and human-like prosody inhibits the effective conveyance of emotions, subtle nuances, and intended emphasis in the speech output. The quest to overcome these limitations and enhance the naturalness of synthesized speech is an active area of research and development. The goal is to augment the user experience, making synthesized speech not just a passive information delivery system, but an engaging medium [5]. Further, the introduction of human-like prosody adds an extra

dimension of expressiveness and nuance to the synthesized speech, making it more relatable and comprehensible for listeners [25].
It helps to convey the intended emphasis, distinguish between different sentence types (such as questions or statements), and express emotions effectively. Furthermore, human-like prosody is particularly crucial in applications where the synthesized speech is intended to engage and communicate with users, such as voice assistants or interactive systems. By infusing the synthesized speech with natural prosody, these systems can provide a more pleasant and interactive user experience, fostering better comprehension, engagement, and overall satisfaction [10].
Our approach differs from others in that we investigate how the model behaves when prominence labels are input at different positions. While many researchers typically provide the model with additional information at the beginning, such as prosodic information, tones, or intonation—depending on what is available to them at the time—we introduce our extra information, the prominence labels, at various locations within the architecture. This allows us to evaluate whether it's more beneficial to introduce prominence as an additional input in the beginning or to integrate prominence labels directly into the attention layer. We also consider creating a new prediction head for prominence labels from the attention layer and back-propagating the loss to determine which method produces superior synthesized speech. While the state-of-the-art model learns directly from the input text, we supplement this by adding prominence labels alongside the input text. This provides additional data for the model to learn the prominence of each word spoken in the dataset.
In our research, we use predicted prominence labels from the text to condition speech generation. Prosody refers to the rhythm, stress, and intonation of speech. In our method, words are assigned prominence labels $\in [0, 2] \subset R$, with 2 indicating the highest level of prominence. This rating system allows for a quantitative representation of the prominence of words in a given text, which can be crucial in understanding the nuances of spoken language [22]. The prominence labels are then utilized in various experimental setups. All setups involve the conditioning of Tacotron 2 [18], a neural network architecture for speech synthesis. The research also introduced an auxiliary task-learning approach for the prediction of the prominence labels. This approach was introduced to inspect the behavior of the model with an extra prediction head, which imitates the Stop token architecture and how it will affect the generated hypothesized speech.
In section [2], we review the existing literature on Tacotron2 and prosodic features addition in the architecture, their impact in making speech more natural, and how other prosodic features are important to enhance the synthesized speech. We begin by introducing our proposed approach [3], which involves conducting experimental variations using predicted prominence labels. Subsequently, we provide details on the experimental setup and the evaluation metrics employed to assess the effectiveness of our approach. In [6] we investigate the learning of different experiments with the help of prominence labels. Finally, in [7] we discuss the implication of our findings and suggest future direction for research in this area.

Chapter 2

Motivation and Related Work

Although text-to-speech (TTS) technology has advanced significantly in recent years, much more can be done to better capture the meaning and style of the original text. The main motive for this research is to enhance the naturalness of generated speech from text. While numerous studies have explored the performance of TTS models, such as Tacotron2 [18], there remains a need to investigate the impact of architectural conditioning [19] and auxiliary task learning [17] on the quality and expressiveness of generated speech. Our research explores the potential of experimental modification in Tacotron2 architecture with prominence labels as inputs.
Other alternative neural network architectures, showed effective performance in TTS fields. Speech2Vec [4] is a sequence-to-sequence framework for learning word embeddings from speech. Learning word embeddings directly from speech enables the model to make use of the semantic information carried by speech that does not exist in plain text. Speech2vec can be seen as the speech version Word2Vec [13]. Their research showed speech contains richer information than text such as prosody, and a machine should be able to make use of this information in order to learn better semantic representations. Their goal was to learn a fixed-length embedding of an audio segment corresponding to the word. The word embedding was able to describe the semantics of the original audio segment.
Another research based on Transformers network [12], used the Tacotron2 framework and incorporated a multi-head attention mechanism, replacing the traditional recurrent neural networks and original attention protocols. This adjustment allowed them the parallel construction of hidden states in the encoder and decoder, significantly improving training efficiency. Multi-head here replaced the original RNN structures in the encoder and decoder as well in the attention. It can split one attention into subspaces so that it can model the frame relationship in multiple different aspects. It is beneficial for learning when sentences are long, as the generated sample sounds smoother. Moreover, the self-attention mechanism facilitates direct connections between inputs at different time intervals, effectively resolving long-range dependency issues. Utilizing phoneme sequences, their Transformer TTS network generates mel spectrograms, which are further processed by a WaveNet vocoder to produce the final audio outputs. Generate audio quality was evaluated by human testers in MOS and CMOS which showed similar results to the base Tacotron2.
The creation of novel speech synthesis algorithms and methodologies, such as neural network-based, has been one of the main areas of attention in TTS research. It has been demonstrated that these methods enhance the generated speech's naturalness and understandability [2, 16].
An early attempt at encoding prosody into TTS systems is demonstrated by Hirschberg and Pierrehumbert (1986) [6]. They presented a set of rules to predict prosodic features such as

intonation and stress. They investigated how variations in intonation can convey important information about discourse structure, referent choice, information status, conceptual contrasts, and subordination relationships. More recent studies have begun to leverage deep learning techniques for this purpose. The Tacotron system, introduced by Wang et al. [20], was one of the first models to demonstrate an end-to-end trainable neural TTS system. It achieved substantial improvements in the naturalness of the speech, but it had no explicit control over prosody.

Y.Lee et al. [11] introduce prosody embeddings for emotional and expressive speech synthesis. The embeddings enable fine-grained control over the speaking style and are integrated into end-to-end synthesis networks. By adjusting the learned prosody features, the pitch and amplitude of the synthesized speech can be changed at both frame and phoneme levels. The authors also propose temporal normalization of prosody embeddings, which improves robustness against speaker variations during prosody transfer tasks.

A study by Ramu et al. [15] delves into the extraction and analysis of both global and local prosodic features from various linguistic units like sentences, words, and syllables, specifically for the task of speech emotion recognition. While global features offer a broad statistical view of prosodic patterns, local features provide insights into the finer, moment-to-moment dynamics of speech. This approach, which considers the positions of words and syllables, emphasizes the depth of information that can be extracted from prosodic features. For our research, this study is informative, we extract insights for prosodic features inclusion into training for TTS models. As it also underscores the importance of considering prosody at various granular levels, from the overarching outlines of a sentence to the specific dynamics of individual syllables. As we aim to enhance Tacotron2 with prominence labels, understanding the multi-layered nature of prosody, as evidenced by [15], offers valuable insights. Their methodology and findings underscore the potential of our approach to capture and reproduce the rich prosodic nuances in synthesized speech. Their exploration of local and global prosodic features as well as positions of words and syllables at different positions and even separately, helped us understand the role of word emphasis/prominence in emotion recognition.

Research by Ramu et al. [15] only focused on the local and global prosodic features of linguistic units. However, research by Yu Zhou et al. [27] introduces a speech emotion recognition system that incorporates both spectral and prosodic features. This research is introduced to effectively present the variations and options to pick and use the prosodic features from various sources and their effects in improving synthesized speech. The spectral features mentioned in this research capture characteristics of speech sounds, while prosodic features reflect elements like pitch and rhythm. By combining these features, their system achieves a significant reduction in emotion recognition errors compared to using each feature type individually.

In a recent study [8], the subtle prosodic nuances of Dutch sarcastic speech were explored. Based on sentence-level analysis, the results revealed specific prosodic indicators for sarcasm, including longer duration, lessened intensity, and diminished vocal loudness when compared to sincere speech. It's interesting to note that the use of pitch and duration varied depending on the sort of statement and the gender of the speaker. In context with our research, these findings underscore the complex and varied nature of prosodic cues in encoding linguistic subtleties. This highlights the potential depth and richness that prominence labels can bring to synthesized speech, enabling it to capture and convey not just the literal meaning but the layered emotional and pragmatic nuances beneath.

Research by Jeremy et al. [1] explores how prosody can be used to identify emotional states during human-computer interactions. Their results show that prosodic models can distinguish

between neutral and emotionally charged utterances with an accuracy comparable to human judgment, particularly those of displeasure. The models continue to be accurate even when they use recognized words rather than precise spoken content. Interestingly, the study also discovered that hyperarticulation is not a valid emotional indicator and language model features lag in predicting emotions. This study shows the informational value of prosody for emotion in speech, albeit in STT in the latter case.

This study highlights the potential of prosodic elements in evoking and expressing emotions for our research. When attempting to increase the emotional resonance of synthesized speech systems, it emphasizes the transformative power of prosody.

Chapter 3

Research Question

The central research question of this thesis is: "How does the output of the Tacotron2 TTS model change specifically in terms of naturalness when integrated with prominence labels?" Prominence labels, derived from highly large-scale datasets, indicate word emphasis. By introducing these labels as input to the Tacotron2 model, we allow the model to better identify the emphasis for each word in the input text. This could lead to more accurate and natural speech synthesis. While the use of prominence labels in speech synthesis has been explored, their integration in the context of architectural modifications and auxiliary task learning in Tacotron2 remains a novel area of investigation.
Our research delves into the behavior of Tacotron2 under different experimental conditions, focusing on the impact of architectural adjustments and auxiliary tasks when prosodic labels are used as the input features in the model.

Hypotheses

- Incorporating predicted prominence as an additional input in the Tacotron2 model will result in a measurable increase in the expressiveness of the synthesized speech as compared to the baseline model without prominence input.
- The direct embedding of prosodic information through concatenation will produce fewer artifacts in the synthesized speech compared to methods that do not embed this information.
- The introduction of prominence prediction as an auxiliary task will lead to faster learning efficiency, as measured by the rate of reduction in prominence loss feedback, compared to the Tacotron2 model without this auxiliary task.

Chapter 4

Method

In this section, the methodology employed to conduct our experiments is described, with a focus on the various modifications applied to the architecture of Tacotron2, wherein prominence labels are utilized as an additional input feature. We anticipate that the synthesized audio will exhibit the desired naturalness. The Tacotron2 model, recognized as a state-of-the-art TTS (Text-to-Speech) architecture, will serve as the baseline model for this research.

4.1 Baseline model: Tacotron2

This first experiment will act as a baseline, where we train the Tacotron2 model with the Ljspeech dataset which contains text with prominence labels.
The Tacotron2[19] is a sequence-to-sequence model comprising two principal components: the encoder and the decoder.

Encoder: The role of the encoder is to convert input text tokens into a rich, high-dimensional latent representation. The encoder does this using several convolutional layers followed by a bidirectional Long Short-Term Memory (LSTM) layer. The input text from the dataset enters the encoder, where it is processed into this high-dimensional latent space. The encoder takes the text portion of the data, which allows it to learn the semantic and syntactic structures of the language. The Encoder's learning of semantic features of the text is somewhat limited and not very deep in comparison to the semantic understanding of dedicated NLP models trained for tasks like question-answering, translation, or summarization. It captures the semantic structure of the language to the extent necessary for high-quality speech synthesis, ensuring that the generated speech reflects the meaning and intent of the input text.

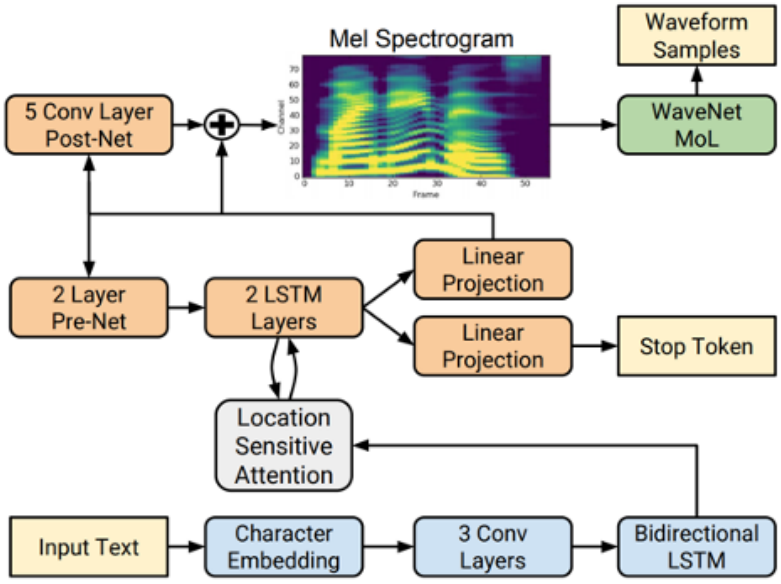


Figure 4.1: Block diagram of the Tacotron2 system architecture.

Attention Mechanism: Before passing the encoded representation to the decoder, the model employs an attention mechanism. This attention mechanism is a crucial component of the Tacotron2 system as it helps to dynamically align the input sequence (text) with the output spectrograms. The attention mechanism considers the entire output of the encoder and determines which parts are most relevant for each decoding step, thereby enabling the model to manage the variability in the alignment between text and corresponding speech.

Decoder: The decoder of Tacotron 2 operates in an autoregressive manner, taking the encoded text and producing mel-spectrogram frames step-by-step. Using a combination of a pre-net, LSTM layers, and a location-sensitive attention mechanism, it ensures accurate alignment between text and generated speech. The process continues until a stop token indicates the end of speech generation.

Post-Net: The post-net in Tacotron 2 refines the mel-spectrogram frame predicted by the decoder. It's a neural network that takes the initial prediction, processes it and produces a correction or refinement. This refined output is then added to the initial prediction to produce a more accurate and higher-quality mel-spectrogram frame.

The Tacotron2 model stands out among text-to-speech models due to its end-to-end nature, and attention mechanism. It's direct learning from text to speech eliminates complex hand-engineered components while enhancing output quality. Furthermore, its ability to accurately process various text inputs, including punctuation and non-standard words, makes it highly applicable in real-world scenarios. These features make Tacotron 2 an ideal choice for tasks demanding efficient, high-quality speech synthesis.

4.2 ENCO-MOD: Encoder modification in Tacotron2

In our adaptation of the Tacotron2 model, we aim to incorporate the prominence labels from the datasets. For data preparation check [5.1]. Prominence labels after being processed and normalized, are converted into vectors. To include prominence labels in the architecture a new embedding layer was introduced for the prosody which matches the size of the character embedding in the architecture, then the prosody embedding layer was concatenated with the character embedding layer to include prominence labels into the input, to match the size of the next layer input layer a new fully-connected projection linear layer was introduced to project concatenated embedding layer into it's original shape. This allows us to maintain architectural consistency. This was done to investigate the effects of conditioning through additional input of prominence labels and architectural overview can be seen in [4.2].

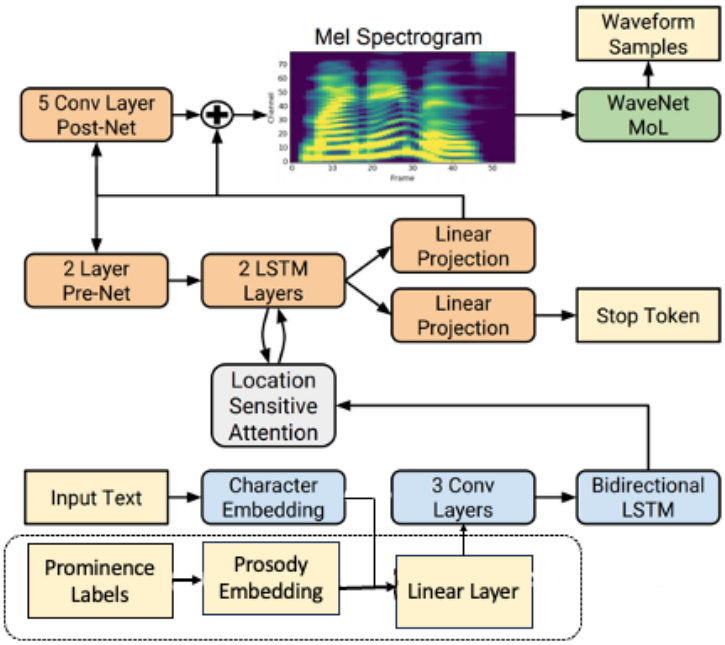


Figure 4.2: Extra input with prominence labels for the encoder layer. The dotted structure represents our architectural additions.

4.3 ATT-MOD: Updated Attention layer input

In this experiment, similar data preparation was used which can be seen in [5.1]. A novel approach was introduced here where the prominence labels are concatenated with the encoder's layer output and passed into the attention module. To execute this experiment, again a prosody embedding layer was introduced and then the results from the encoder layer were concatenated and later passed through a fully connected projection linear layer to ensure the resultant results were in a similar shape for the attention's output.
By concatenating prominence labels with the encoder's output, the model is afforded a more direct route to utilize prosodic information during the subsequent stages of processing. This is fundamentally different from the traditional Tacotron 2 approach, where the model must infer prosodic properties indirectly from the textual input.

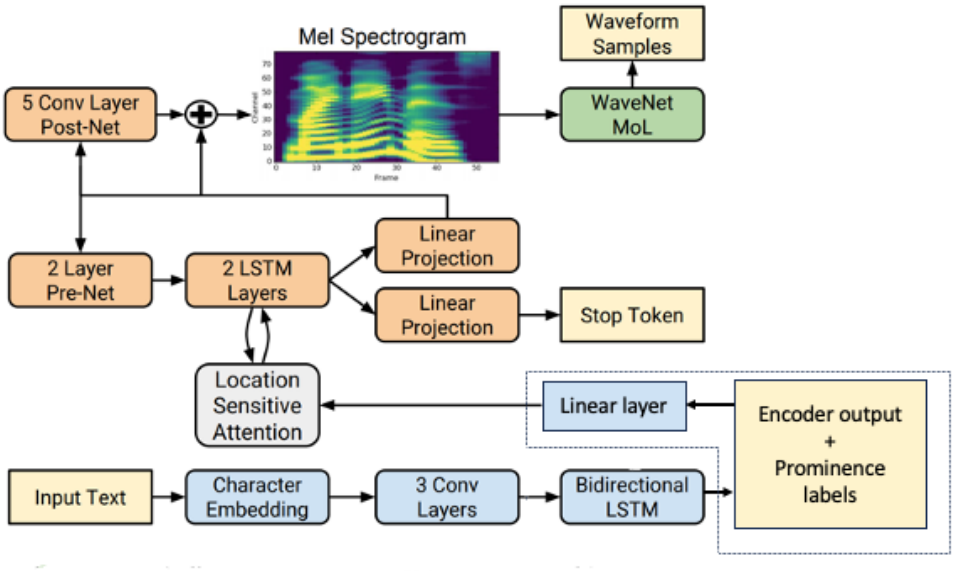


Figure 4.3: Updated output of encoder layer with prominence labels. The dotted structure represents our architectural additions.

After modifying the encoder output, this new data is subsequently passed into the attention mechanism of Tacotron2 [4.3]. The attention mechanism plays a key role in aligning input textual data with the corresponding portions of the generated speech. By feeding prominence-enhanced encoder outputs into the attention mechanism, it is anticipated that this will allow the model to learn more accurately the relationship between text and corresponding speech prosody.

4.4 PRO-MOD: Predicting loss on attention layer

In this study, we implement an auxiliary task-learning approach within the Tacotron2 model architecture. The goal of this experiment is to study if predicting prominence labels also helps in generating better audio in terms of prosody.

To implement this, a novel approach was introduced where we propose an auxiliary prominence loss prediction head in the model. This prominence prediction head is a small fully-connected layer connected to the output of the attention module within the Tacotron2's architecture. It predicts the prominence of the current attention readout. This is done to force attention output to also contain information that is relevant to predict the prominence of the current to be decoded frames. The loss is analogous to the stop token loss, but based on the prediction of the prominence labels. The prominence prediction head outputs a sequence of predicted prominence labels, the disparity between these predicted labels and the actual prominence labels forms the prominence loss. The loss(Cross-Entropy Loss) provides a measure of how far the model's prominence predictions deviate from the true prominence features in the training data. The alignment of predicted frame-based prosody labels with the true text-based prosody labels is managed in the loss function. For the purpose of training, the true prosody labels are expanded (repeated) to match the length of the predicted labels, ensuring that each frame has an associated prosody label. In the inference phase, the final prosody label for each frame is typically determined by taking the argmax of the logits produced by the model and selecting the class with the highest predicted value. However, the method of expanding true labels during training is a direct approach to handle different lengths between predictions and targets.

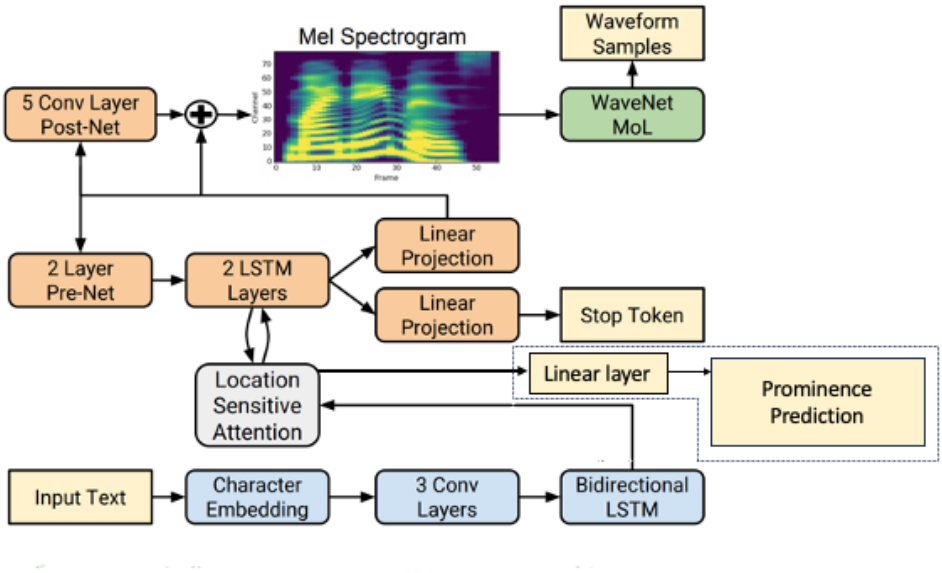


Figure 4.4: Auxiliary task: Adding Prominence Loss Prediction Head. The dotted structure represents our architectural additions.

The calculated prominence loss is then backpropagated through the model during training, adjusting the model's parameters to reduce this prominence loss in future predictions. By integrating this prominence loss into the overall loss of the Tacotron2 model, the model is encouraged to generate speech that not only aligns with the text but also exhibits realistic prominence patterns.

4.5 Vocoder: Waveglow

In the field of speech synthesis and audio processing, a vocoder serves as a key element that facilitates the conversion of Mel-frequency cepstral coefficients (MFCCs), or Mel spectrograms, back into waveform signals, a process often denoted as Mel-to-Wave synthesis. Utilizing complex algorithms and neural network architectures, it reconstructs the time-domain waveform from the spectral representation, maintaining a thorough balance to preserve the details of the original audio signal. For more details on Waveglow see [14].

4.6 Measures

4.6.1 Loss functions

The loss functions being used in the Tacotron2 model are described below. In our Tacotron 2 implementation, we chose to evenly weight the losses to ensure balanced importance between generating accurate spectrograms and determining the sequence's end. Empirical evaluations indicated that this configuration produced optimal performance. Additionally, even weighting prevented bias towards any specific objective, maintaining the model's focus on both the quality of speech synthesis and its natural termination.

Mel Spectrogram Loss (Mel Loss)

Given: The Mel loss is computed as the sum of the MSE loss between the prediction before PostNet and the target Mel spectrogram, and the MSE loss between the prediction after PostNet and the target Mel spectrogram.

- y as the ground truth Mel spectrogram.
- $\hat{y}_1$ as the predicted Mel spectrogram before the PostNet.
- $\hat{y}_2$ as the predicted Mel spectrogram after the PostNet.

The Mel loss L_{mel} is:

$$L_{mel} = \| y - \hat{y}_1 \|_2^2 + \| y - \hat{y}_2 \|_2^2 \tag{4.1}$$

Gate Loss (Stop Token Loss)

Given: The gate loss is calculated using binary cross-entropy with logit loss between the predicted gate outputs and the target gate outputs.

- g as the ground truth stop token (0 or 1).
- $\hat{g}$ as the predicted stop token (logit output, before applying a sigmoid activation).

The gate loss L_{gate} is:

$$L_{gate} = -g \log(\sigma(\hat{g})) - (1 - g) \log(1 - \sigma(\hat{g})) \tag{4.2}$$

where $\sigma(.)$ represents the sigmoid activation function.

Prominence Loss

Given: The prominence loss is computed using the cross-entropy loss between the predicted prosody values and the prosody target values.

- p as the prosody target values.
- $\hat{p}$ as the predicted prosody values.
- The p_i refers to the prosody target value for the *i-th* element in the sequence. Similarly, $\hat{p_i}$, refers to the predicted prosody value for the *i-th* element.

The prominence loss $L_{prominence}$ is:

$$L_{prominence} = -\sum_{i} p_i \log(\hat{p}_i) + (1 - p_i) \log(1 - \hat{p}_i) \tag{4.3}$$

Chapter 5

Experimental Setup

5.1 Dataset

In the initial phase of our experiments, we utilized the LibriTTS dataset [26], which encompasses similar features and prominence labels for words, deriving from a multispeaker platform with 2,456 speakers. The prominence labels in LibriTTS were appended by the Helsinki Prosody corpus [22], utilizing the Continuous Wavelet Transform annotation method [21] and Wavelet prosody analyzer toolkit.
Simultaneously, we integrated the LJ Speech dataset [7], consisting of 3,100 short audio clips from a single speaker reading passages from 7 non-fiction books, into our experiments. Recognizing the absence of prominence labels in this dataset, we formulated a method to introduce them. In the dataset preparation, a heuristic labeling approach was applied to designate words with prominence labels: "low" (0), "medium" (1), and "high" (2), based on their lexical and syntactic roles. Common function words and certain auxiliary/modal verbs were typically assigned low and medium prominence, respectively, while other content-rich words were labeled high prominence. Punctuation marks were assigned a distinct label ('NA') and treated separately to maintain sentence structure. Here's an example of the label generation for the LjSpeech dataset,

Sentence: "The quick brown fox jumps over the lazy dog."

Processing:

- Common words like "the" might be considered of low prominence: label 0.
- Words like "jumps" and "over" could be considered of medium prominence due to their functional roles: label 1.
- The remaining words, which tend to carry significant meaning or are content words, would be considered of high prominence: label 2.

Labeled Sentence: "The 0 quick 2 brown 2 fox 2 jumps 1 over 1 the 0 lazy 2 dog 2."

Data Preparation for training

To prepare the dataset, we obtained the text and their corresponding prominence labels from the text files. They have extracted the prominence labels from the audio files corresponding to their sentences. So the final dataset looked like this: "Audio file — Sentence with their prominence labels". For the first batch of experimentation, we used the Libritts dataset, and for the second batch of experimentation, we used the LjSpeech dataset.

5.2 Training

The models were trained using a dataset of sentences with prominence labels with their Mel Spectrogram representations of the audio signals. The training process involved the following steps:

- Dataset: Dataset was divided for training, testing, and validation in the ratio of 8:1:1.
- Optimizer and Learning Rate: The model uses the Adam optimizer with a learning rate of 1e-3 and a weight decay of 1e-6.
- Training Procedure: The model was trained using small-batch stochastic gradient descent. A batch size of 4 and 32 were used. The model was trained for a total of 10 epochs which account for 120k iterations for batch size 4 and 500 epochs which account for 180k iterations for batch size 32. During training, the backpropagation algorithm computed the gradients of the loss function with respect to the model parameters, and the optimizer updated the parameters accordingly.
- Regularization: During training, the model employs L2 regularization (also known as weight decay) set at a rate of 1e-6 to prevent overfitting, and also uses gradient clipping with a threshold of 1.0 to mitigate the effects of exploding gradients.
- Monitoring and Evaluation: Throughout the training process, the model undergoes evaluation against a validation set every 1000 iterations, and corresponding checkpoints of its state are saved for potential resumption or analysis.

Chapter 6

Results

Several observations were made based on the results we obtained. To establish a performance baseline, we also added and presented results with the baseline Tacotron2 model.
ENCO-MOD: Encoder modification in Tacotron2 [4.2]
ATT-MOD: Updated Attention layer input [4.3]
PRO-MOD: Predicting loss on attention layer [4.4]

In our systematic analysis, we leveraged violin plots to visualize and interpret the distribution of Mel and Gate losses across different experimental setups which can be seen in Figures. Violin plots for the Gate loss range from [0.00-0.14] and for Mel loss it ranges from [0.0-1.2] [6.1, 6.2, 6.3, and 6.4]

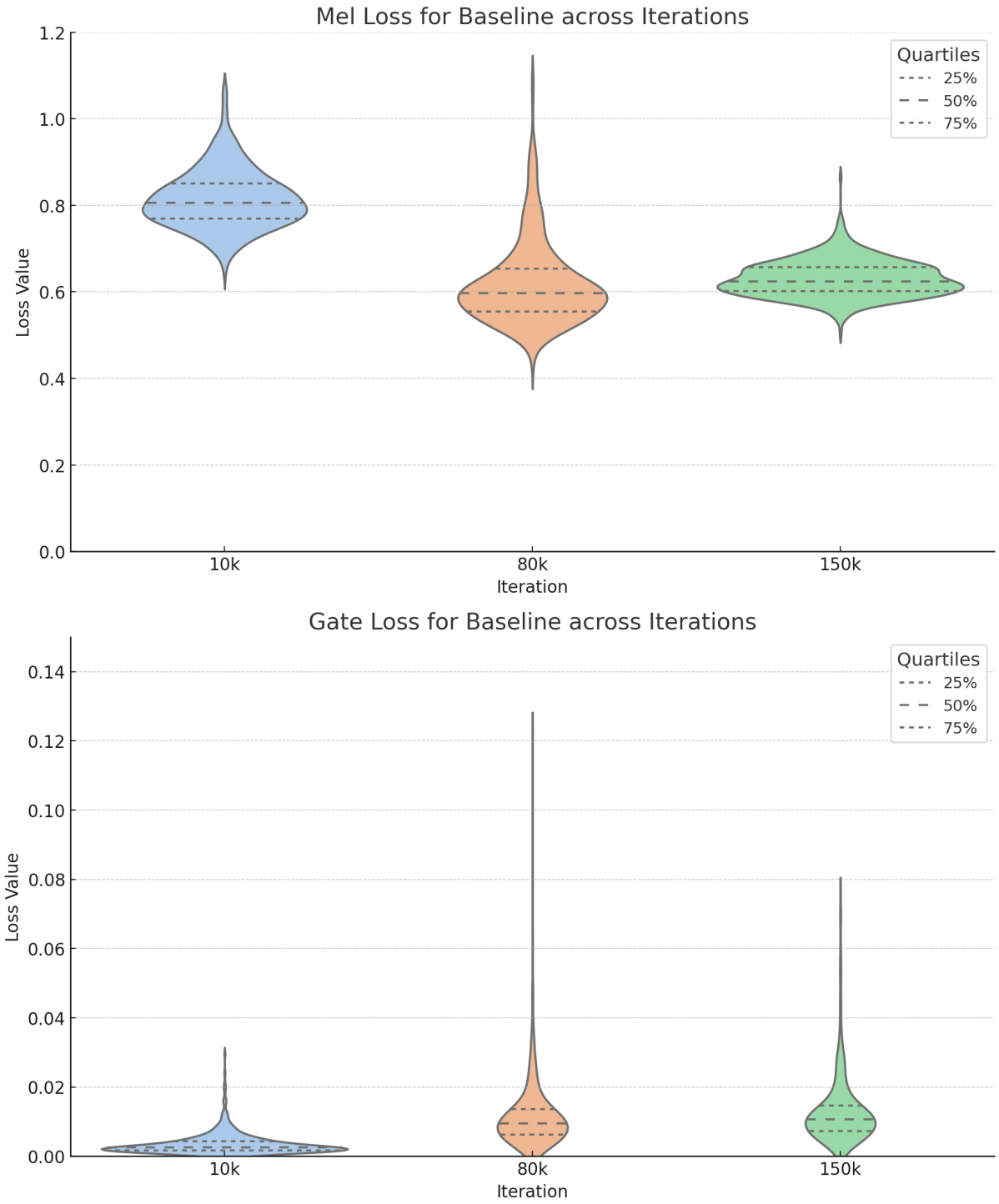


Figure 6.1: Comparison violin plot of Gate and Mel loss for baseline on 3 different iterations checkpoints tested on the test dataset.

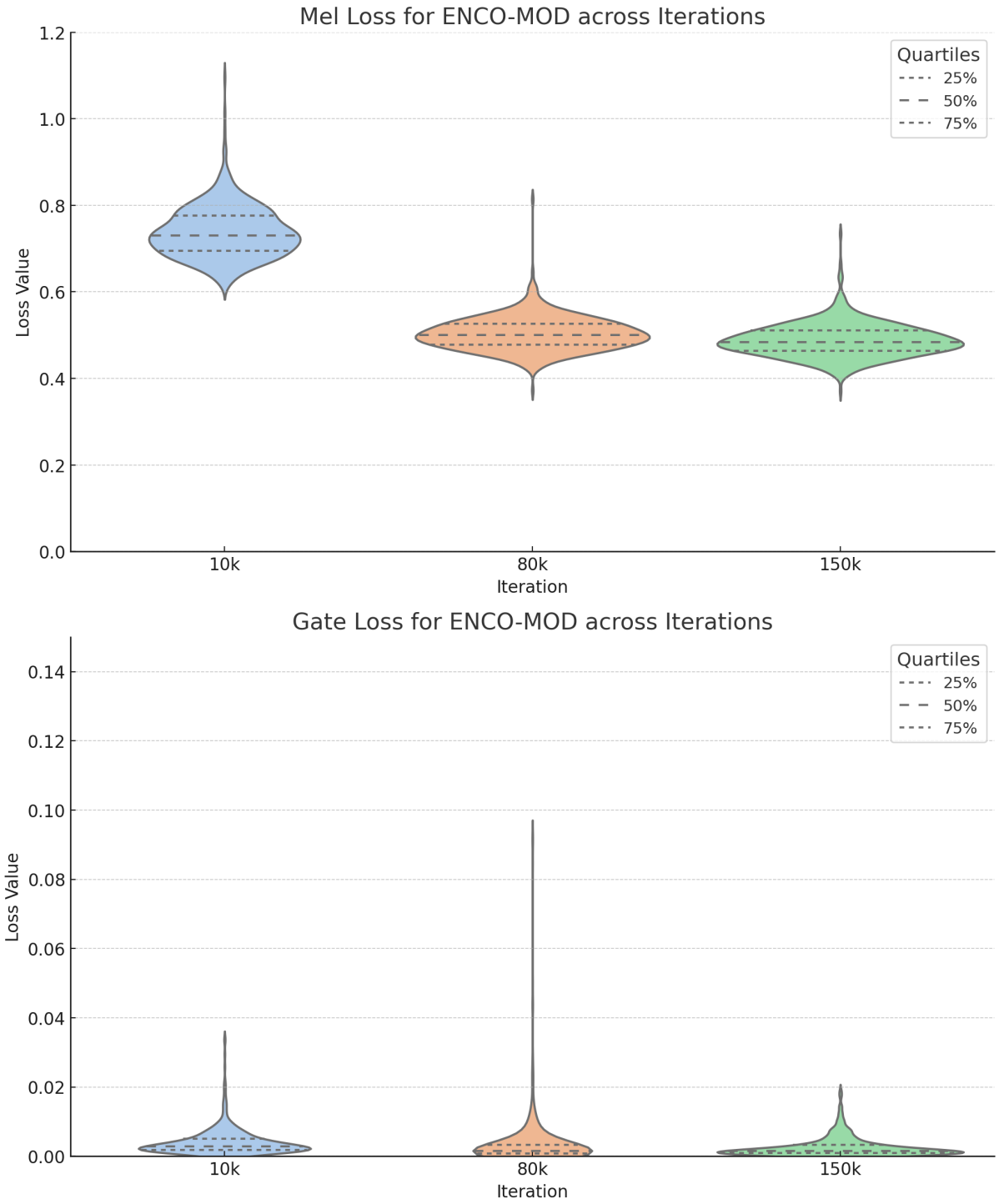


Figure 6.2: Comparison violin plot of Gate and Mel loss for ENCO-MOD on 3 different iterations checkpoints tested on the test dataset.

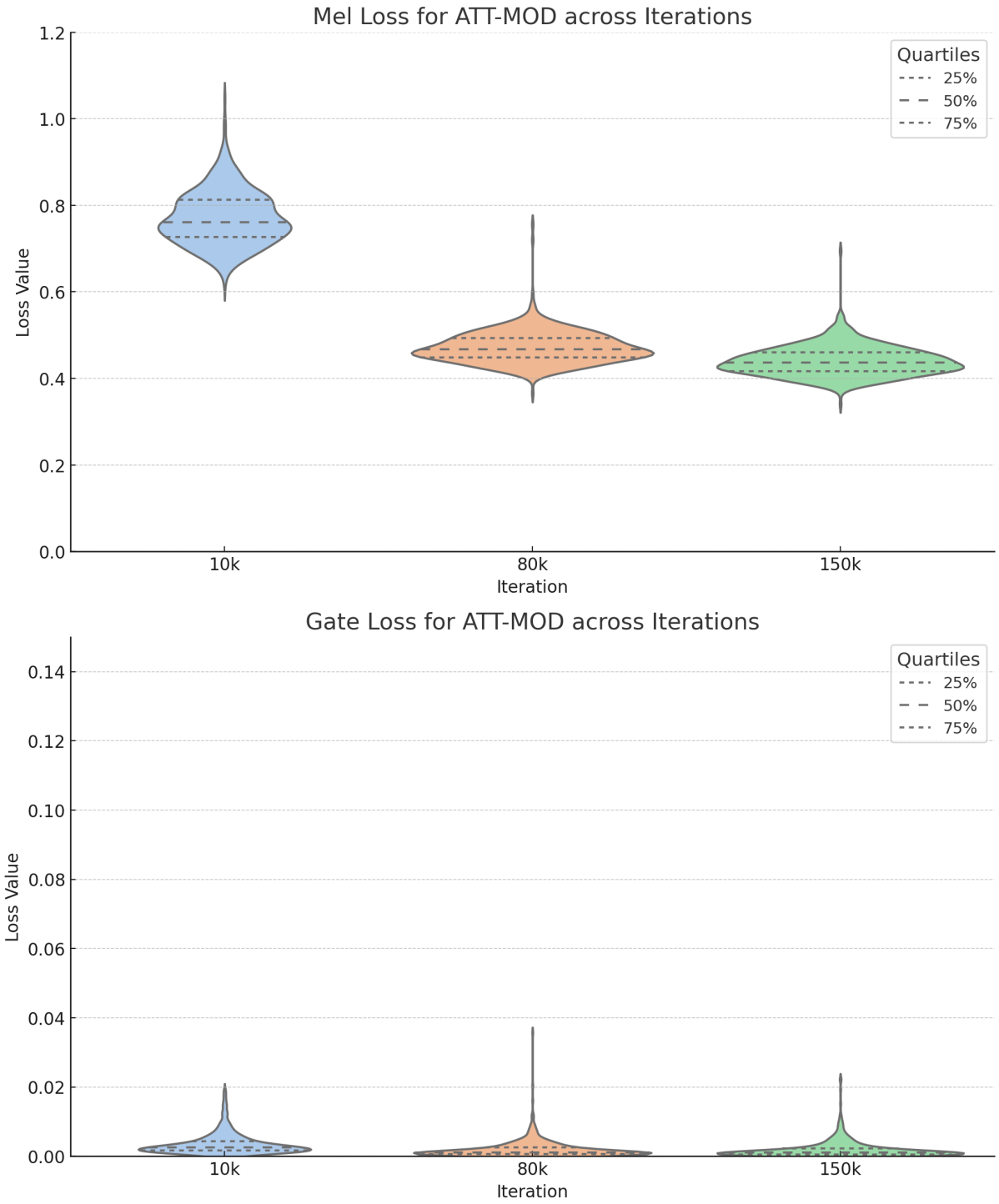


Figure 6.3: Comparison violin plot of Gate and Mel loss for ATT-MOD on 3 different iterations checkpoints tested on the test dataset.

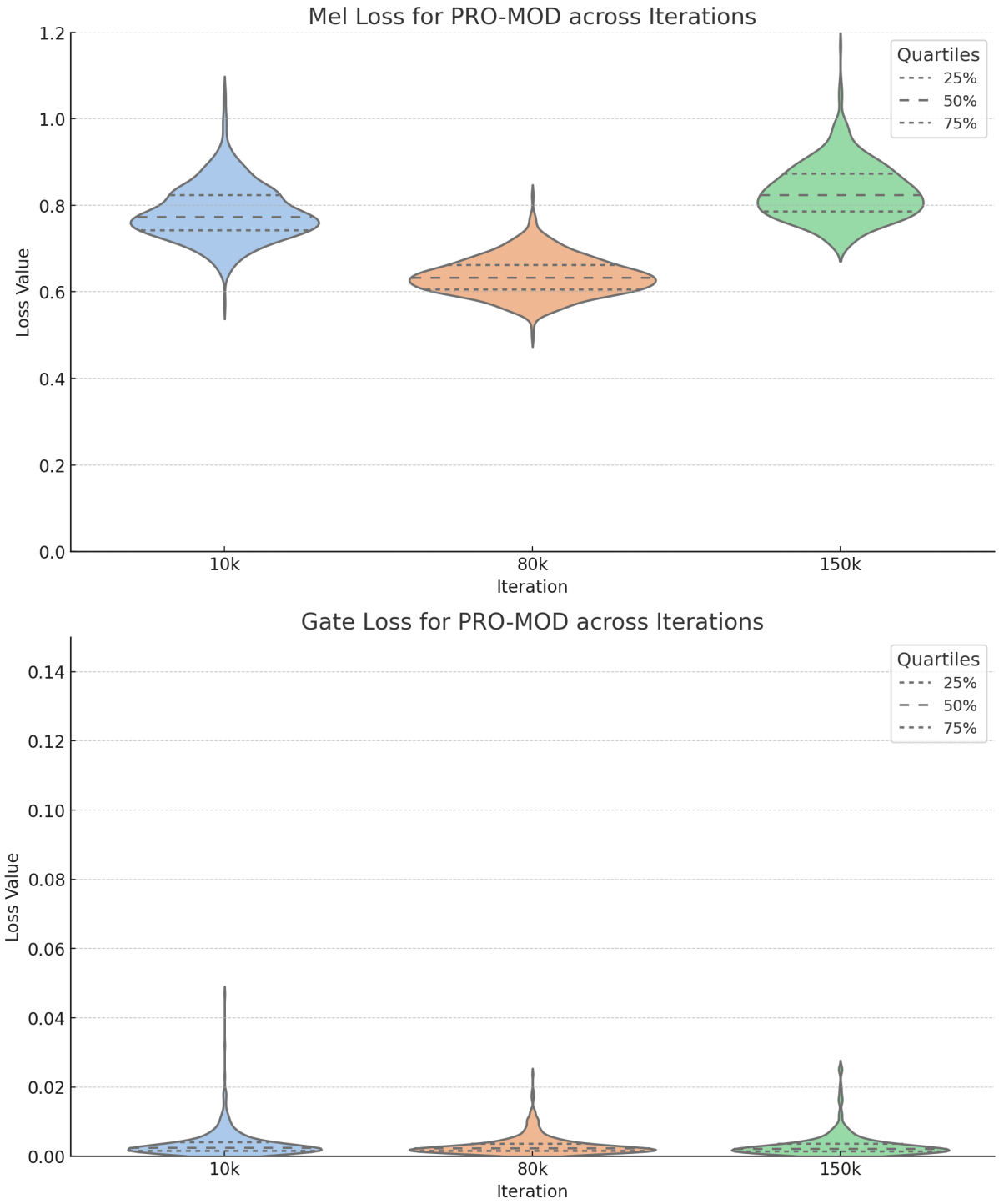


Figure 6.4: Comparison violin plot of Gate and Mel loss for PRO-MOD on 3 different iterations checkpoints tested on the test dataset.

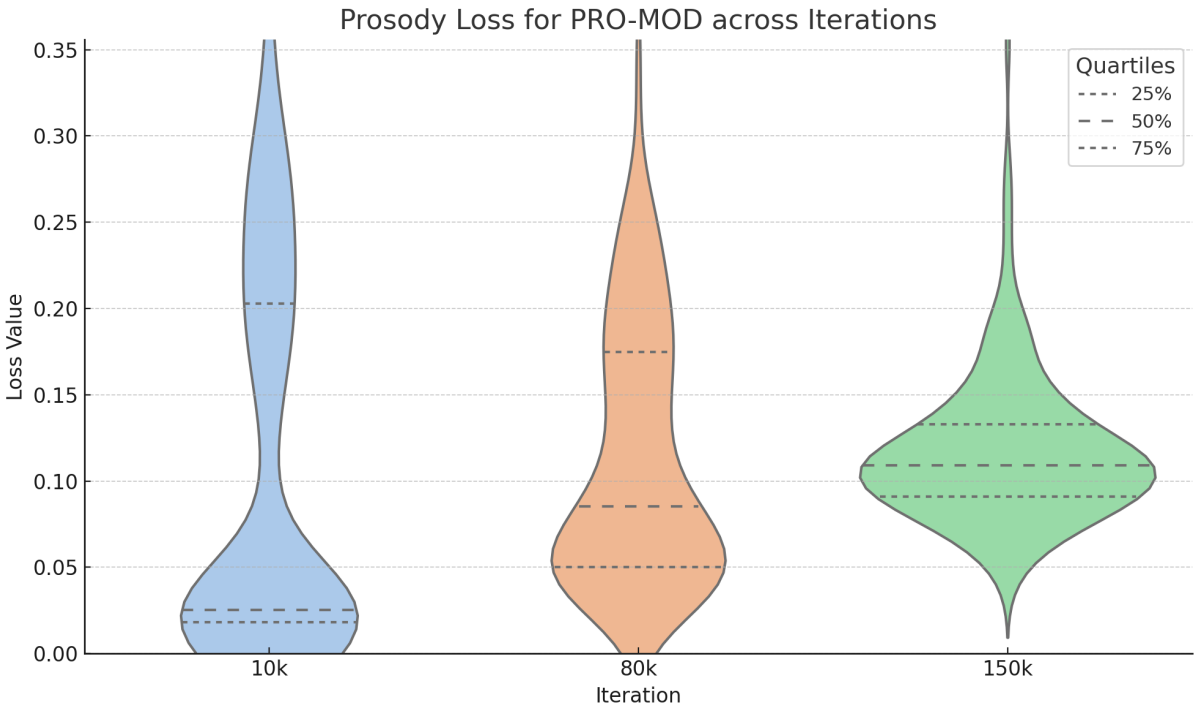


Figure 6.5: Comparison violin plot of Prosody loss for PRO-MOD on 3 different iterations checkpoints tested on the test dataset.

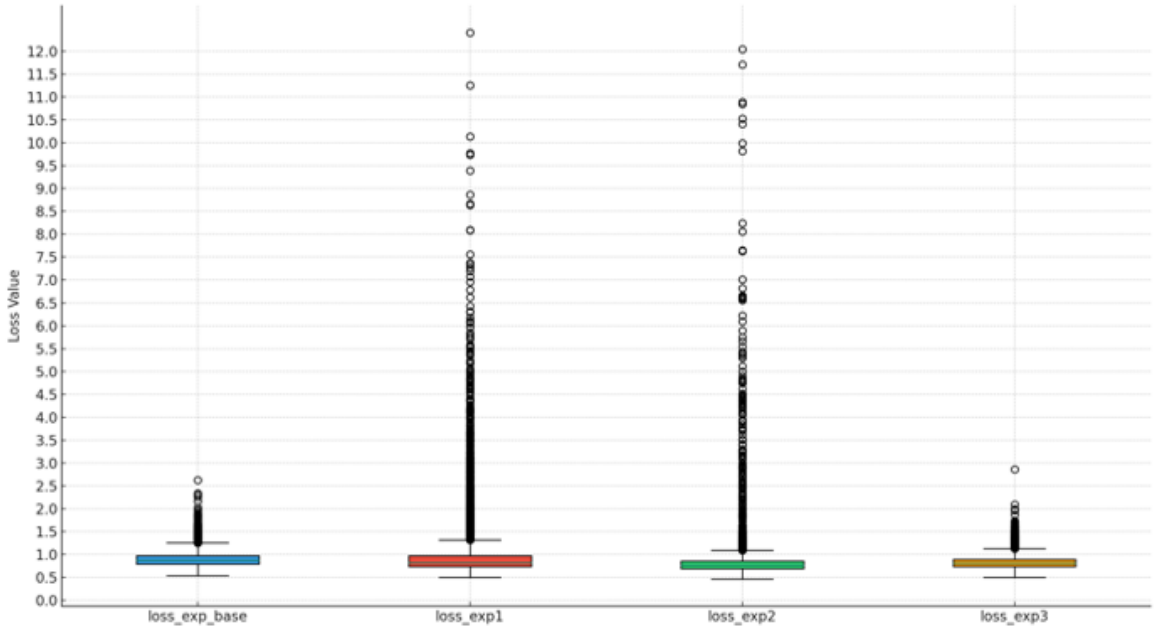


Figure 6.6: Box plot for combined loss for all three experiments with a baseline on the test dataset. Exp1 here is ENCO-MOD, Exp2 is ATT-MOD, and Exp3 is PRO-MOD.

6.1 Analytical Overview on violin plots

6.1.1 ENCO-MOD Analysis

[6.2]

Mel Loss Overview

ENCO-MOD vs. Baseline: ENCO-MOD consistently shows a lower median Mel Loss across all iterations compared to the baseline, indicating a more accurate model on average in synthesizing Mel spectrograms.
Key Observations: However, the distribution width (representing variability in predictions) fluctuates across iterations, indicating periods of both stable and variable predictive performance.

Gate Loss Overview

ENCO-MOD vs. Baseline: The Gate Loss in ENCO-MOD is notably higher than in the baseline across all iterations, suggesting less accurate predictions in phoneme durations.
Key Observations: The consistently broader distributions also suggest higher variability in prediction performance across different input samples.

Iteration Insights

Mel Loss tends to decrease as iterations progress, showing model learning and optimization. Gate Loss fluctuates across iterations, indicating inconsistent performance in predicting phoneme durations.

Summary: ENCO-MOD

ENCO-MOD demonstrates commendable performance in reducing Mel Loss, indicating proficient Mel spectrogram predictions. However, the high and variable Gate Loss signals potential inconsistencies and challenges in accurately predicting phoneme durations, which might impact the synthesized speech's naturalness and coherence.

6.1.2 ATT-MOD Analysis

[6.3]

Mel Loss Overview

ATT-MOD vs. Baseline: ATT-MOD generally exhibits higher Mel Loss values compared to ENCO-MOD but is still lower than the baseline, placing its performance in a middle ground between the two.

Gate Loss Overview

ATT-MOD vs. Baseline: The Gate Loss in ATT-MOD, while lower than ENCO-MOD, still surpasses the baseline, indicating an area that might require further investigation and optimization.

Iteration Insights

Mel Loss shows a general decrease with increasing iterations. Gate Loss, conversely, trends upwards, signaling potential challenges in optimizing gate predictions as training progresses.

Summary: ATT-MOD

ATT-MOD navigates a middle path, performing better than the baseline but not as proficiently as ENCO-MOD in Mel Loss. Gate Loss remains a consistent challenge, warranting further exploration into optimizing gate predictions.

6.1.3 PRO-MOD Analysis

[6.4]

Mel Loss Overview

PRO-MOD vs. Baseline: Mel Loss in PRO-MOD mirrors that of ATT-MOD, indicating similar average performances but showcasing a wider distribution, revealing more variability in predictions.

Gate Loss Overview

PRO-MOD vs. Baseline: Gate Loss in PRO-MOD is notably higher than both ENCO-MOD and ATT-MOD and also the baseline, signaling a potential area of concern and scope for improvement.

Iteration Insights

Mel Loss generally decreases, showcasing model optimization. Gate Loss displays fluctuations, highlighting periods of varied performance in gate predictions.

Summary: PRO-MOD

PRO-MOD aligns closely with ATT-MOD in Mel Loss but introduces more variability, potentially signaling less consistency in predictions. The Gate Loss, being the highest among the three experiments, pinpoints a critical area that might require focused attention and optimization in future iterations.

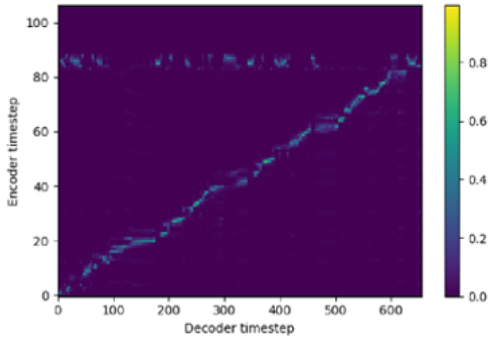


Figure 6.7: Alignment observed during baseline experiment with batch size 32 and around 38k steps it started showing early signs of the alignment, with Ljspeech dataset.

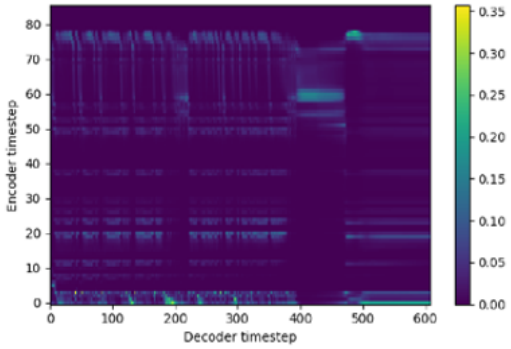


Figure 6.8: No alignment observed during baseline experiment with batch size 32, till 150k steps with Libritts dataset.

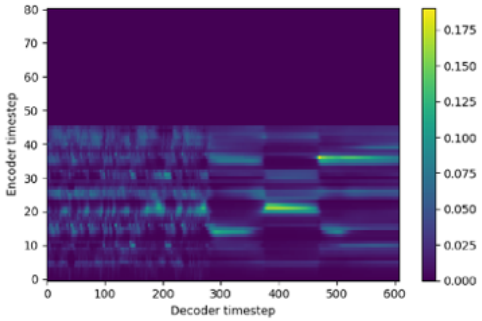
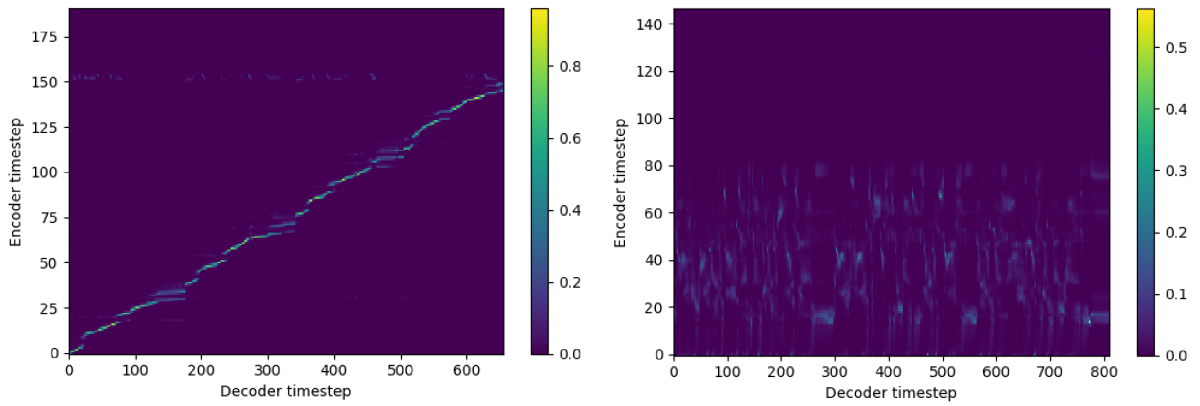
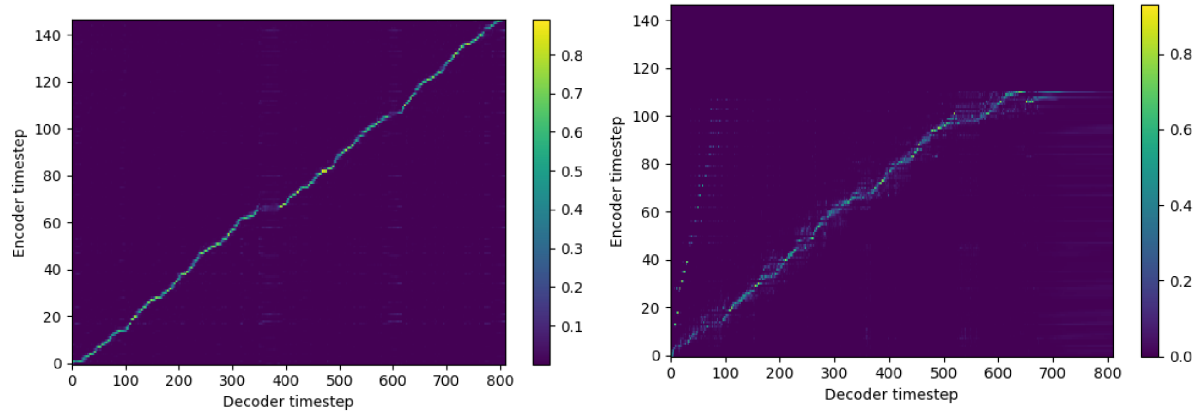


Figure 6.9: No alignment observed during the second experiment with batch size 32, till 320k steps with Libritts dataset.



(a) Baseline

(b) ENCO-MOD



(c) ATT-MOD

(d) PRO-MOD

Figure 6.10: Alignment observed in Baseline, ATT-MOD, and PRO-MOD. ENCO-MOD failed to attain alignment. Shown experiments are trained on the Ljspeech dataset till 150k iterations.

6.2 Discussion

The observed outcomes from training show great potential. Initially, training was performed for all 4 experiments including baseline till 100k steps on the batch size of 4 with our prominence labeled data. Later during the inference, we found out that the generated audio was noisy. So we trained Vocoder on our dataset so that it could get accustomed to longer sentences and audio files. However, our experiments didn't show any alignment; even the baseline model showed no alignment till 100k steps during inference. Later we tested our findings on a different dataset, the Ljspeech [7] dataset. This dataset showed us that when the model is trained with batch size 32 it gets alignment around 38k steps, as shown in [6.7]. While batch size 4 failed to show any alignment till 120k steps in iteration. This finding leads us to investigate more on how we can tune the hyperparameters. We found out that batch size and learning rate are correlated to each other so, we tried another experiment with the Ljspeech dataset again, but this time with batch size 4, and the learning rate was reduced 8 times as well. But this failed to show the alignment during training till 100k steps. This made us realize that a higher batch size is required for the model to learn the alignment in the early stage of the training. As we didn't take this information into account during initial training, we modified our prominence

labels dataset by limiting only text of a certain length to be used for a batch size of 32. Then we ran the training again but only for the baseline and for our second experiment to test our findings. While training, we observed that for the baseline no alignment was observed till 130k steps, and for the second experiment no alignment even at 320k steps as shown in Figures [6.8][6.9]. These results failed our findings based on the batch size and the learning rate. Later it was found that our prominence labeled dataset(LibriTTS) is actually a multispeaker dataset that contains nearly 2456 speakers and Tacotron2 is a single-speaker TTS model. This made us realize that this is the reason our model is not achieving the alignment because for one speaker Tacotron2 was taking 40k-50k steps to get perfect alignment and for the 2456 speakers it would approximately take $2456 * 50k$ steps. So hypothetically a very long training process would have been required in the case of this multispeaker problem and maybe it could attain the alignment as well with a longer training process.

To overcome this challenge, we trained our models on the Ljspeech dataset with prominence labels, which is originally a single-speaker dataset and has already proven earlier in the testing that it attains alignment around 38k steps. During training, we only trained our experiments to 150k steps to check if our experiments were indeed learning the alignments. Notably, with the exception of ENCO-MOD, alignment was attained by each experiment up to 150k steps. Moreover, ATT-MOD exhibited the most favorable alignment during training, while PRO-MOD displayed consistent growth in alignment learning, suggesting that extended training duration could prove advantageous for PRO-MOD by potentially showcasing the efficacy of adding prosody loss in the Tacotron2 architecture. The observed alignment plots are presented in [6.10].
In contrast, during the inference stage, the audio generated by PRO-MOD was most akin to that produced by the Baseline model, though it is early to claim that extra prosody loss head helped the model to put more emphasis on words based on prominence labels because of the early stage of training. Although ATT-MOD demonstrated consistent alignment throughout training, it unexpectedly failed to generate coherent speech during inference. Similarly, ENCO-MOD was unable to generate coherent speech, a result of not achieving alignment within 150k training steps. Consequently, while PRO-MOD yielded the best-generated speech when compared to the others, a lengthier training duration is requisite to truly ascertain its capabilities.

Chapter 7

Conclusion and Further Research

This research embarked on a journey to explore the enhancement of synthesized speech's naturalness, particularly through the incorporation of prominence labels into the Tacotron2 architecture. Despite the methodological insights gained, the anticipated outcomes, especially in terms of achieving consistent and meaningful improvements in speech emphasis and word prominence, were not fully realized in the conducted experiments.

Our journey confronted substantial challenges, notably alignment difficulties encountered when utilizing a multi-speaker dataset, which yielded inconclusive and non-generalizable outcomes. A pivot to a single-speaker dataset did illuminate potential paths forward, facilitating alignment achievement in specific experimental setups. Nevertheless, the findings must be contextualized within the limitations of the particular datasets and experimental configurations utilized.

Notably, while ATT-MOD and PRO-MOD demonstrated alignment and indicated that our architectural modifications were fundamentally compatible with the original architecture, their practical application in synthesizing speech with enhanced prominence was limited. PRO-MOD did produce synthesized speech that bore some degree of emphasis, yet without a significant uplift in naturalness or a robust confirmation of the utility of the additional prosody loss head in the architecture, especially considering the early stage of training.

These findings, while providing a foundation, underscore the necessity for caution in deriving broad conclusions. The insights gained are tightly bound to the specific setups and conditions of our experiments and should not be extrapolated to wider contexts without further validation.

Future research will delve deeper, exploring extended training durations and further model refinement, capitalizing on the initial findings gleaned from the single-speaker dataset experiments. A significant focal point will be navigating the complexities of achieving alignment in a multi-speaker dataset, possibly exploring the incorporation of speaker embeddings, and further probing into the optimal integration points and methods for prominence labels in the Tacotron2 architecture.

Bibliography

[1] Jeremy Ang, Rajdip Dhillon, Ashley Krupski, Elizabeth Shriberg, and Andreas Stolcke. Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In *INTERSPEECH*, pages 2037–2040. Citeseer, 2002.

[2] Sercan Ö. Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. Deep voice: Real-time neural text-to-speech. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 195–204. PMLR, 06–11 Aug 2017.

[3] Taejun Bak, Junmo Lee, Hanbin Bae, Jinhyeok Yang, Jae-Sung Bae, and Young-Sun Joo. Avocodo: Generative adversarial network for artifact-free vocoder, 2022.

[4] Yu-An Chung and James Glass. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. *arXiv preprint arXiv:1803.08976*, 2018.

[5] Tobias Cornille, Fengna Wang, and Jessa Bekker. Interactive multi-level prosody control for expressive speech synthesis. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8312–8316, 2022.

[6] Julia Hirschberg and Janet Pierrehumbert. The intonational structuring of discourse. In *24th Annual Meeting of the Association for Computational Linguistics*, pages 136–144, 1986.

[7] Keith Ito and Linda Johnson. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/, 2017.

[8] Nelleke Jansen, Aoju Chen, et al. Prosodic encoding of sarcasm at the sentence level in dutch. In *10th International Conference on Speech Prosody, Tokyo*, pages 25–28, 2020.

[9] Karolina Kuligowska, Paweł Kisielewicz, and Aleksandra Włodarz. Speech synthesis systems: Disadvantages and limitations. *International Journal of Engineering and Technology(UAE)*, 7:234–239, 05 2018.

[10] Karolina Kuligowska, Paweł Kisielewicz, and Aleksandra Włodarz. Speech synthesis systems: Disadvantages and limitations. *International Journal of Engineering and Technology(UAE)*, 7:234–239, 05 2018.

[11] Younggun Lee and Taesu Kim. Robust and fine-grained prosody control of end-to-end speech synthesis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5911–5915, 2019.

[12] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6706–6713, 2019.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[14] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[15] K Sreenivasa Rao, Shashidhar G Koolagudi, and Ramu Reddy Vempada. Emotion recognition from speech using global and local prosodic features. *International journal of speech technology*, 16:143–160, 2013.

[16] Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari. Statistical parametric speech synthesis incorporating generative adversarial networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):84–96, 2018.

[17] Mayank Sharma. Multi-lingual multi-task speech emotion recognition using wav2vec 2.0. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6907–6911, 2022.

[18] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, Rif A. Saurous, Yannis Agiomvrgiannakis, and Yonghui Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783, 2018.

[19] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, Rif A. Saurous, Yannis Agiomvrgiannakis, and Yonghui Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783, 2018.

[20] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pages 4693–4702. PMLR, 2018.

[21] Antti Suni, Juraj Šimko, Daniel Aalto, and Martti Vainio. Hierarchical representation and estimation of prosody using continuous wavelet transform. *Computer Speech Language*, 45:123–136, 2017.

[22] Aarne Talman, Antti Suni, Hande Celikkanat, Sofoklis Kakouros, Jörg Tiedemann, and Martti Vainio. Predicting prosodic prominence from text with pre-trained contextualized word representations. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 281–290, Turku, Finland, September–October 2019. Linköping University Electronic Press.

[23] Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. A survey on neural speech synthesis, 2021.

[24] Paul Taylor. *Text-to-Speech Synthesis*. Cambridge University Press, 2009.

[25] Noé Tits, Kevin El Haddad, and Thierry Dutoit. Exploring transfer learning for low resource emotional tts. In *Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys) Volume 1*, pages 52–60. Springer, 2020.

[26] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

[27] Yu Zhou, Yanqing Sun, Jianping Zhang, and Yonghong Yan. Speech emotion recognition using both spectral and prosodic features. In *2009 International Conference on Information Engineering and Computer Science*, pages 1–4, 2009.

Appendix A

Usage of ChatGPT

ChatGPT has been a helpful device in my project. I basically involved ChatGPT as a composing help, explicitly for help in organizing the segments. In a request to get a decent layout for the segments where I present my data in a sensible way, ChatGPT was valuable. For instance, on account of the Abstract, I gave ChatGPT all the data regarding my task, including the primary goal and the aftereffects of my review. I was given an example abstract segment with a decent progression of data that summed up the task. The example structure was as per the following - it began with what is the principal objective of the paper, which was trailed by the four tacotron2 models used. Then it referenced the commotion types, the assessment measurements utilized, and the aftereffects of every one of the models. I involved this example as a gauge to further form my Abstract segment. Since my emphasis was more on getting a perfect voice output with prosody, I abbreviated the data about the models. Essentially, for the Conclusion area, I utilized ChatGPT to characterize a layout for the information presentation. I held the construction yet gave the data that was applicable to my thesis.