IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.1120000

Joint Speech-Text Embeddings for Multitask Speech Processing

MICHAEL GIAN GONZALES[1], (Member, IEEE), PETER CORCORAN[2], (Fellow, IEEE), NAOMI HARTE[3], (Member, IEEE) and MICHAEL SCHUKAT[1], (Member, IEEE)

[1]School of Computer Science, College of Science and Engineering, University of Galway, Galway, H91 TK33 Ireland
[2]Department of Electronic Engineering, College of Science and Engineering, University of Galway, Galway, H91 TK33 Ireland
[3]School of Engineering, Trinity College Dublin, Dublin 2, D02 PN40 Ireland

Corresponding author: Michael Gian Gonzales (e-mail: M.Gonzales1@universityofgalway.ie)

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224.

ABSTRACT Devices that use speech as the communication medium between human and computer have been emerging for the past few years. The technologies behind this interface are called Automatic Speech Recognition (ASR) and Text-to-Speech (TTS). The two are distinct fields in speech signal processing that have independently made great strides in recent years. This paper proposes an architecture that takes advantage of the two modalities present in ASR and TTS, speech and text, while simultaneously training three tasks, adding speaker recognition to the underlying ASR and TTS tasks. This architecture not only reduces the memory footprint required to run all tasks, but also has performance comparable to single-task models. The dataset used to train and evaluate the model is the CSTR VCTK Corpus. Results show a 97.64% accuracy in the speaker recognition task, word and character error rates of 18.18% and 7.95% for the ASR task, a mel cepstral distortion of 4.31 and two predicted MOS of 2.98 and 3.28 for the TTS task. While voice conversion is not part of the training tasks, the architecture is capable of doing this and was evaluated to have 5.22, 2.98, and 2.73 for mel cepstral distortion and predicted MOS, respectively.

INDEX TERMS Automatic speech recognition, joint speech-text, text-to-speech, speaker recognition, speech processing, voice conversion.

I. INTRODUCTION

Emerging smart devices, such as smart speakers, phones, and tablets, often use speech as the communication medium with the users. Some companies are starting to use conversational AI agents in their hotlines for customer service, replacing the traditional interactive voice response systems [1], [2]. The underlying technologies that enable the use of speech between human and computer communications are called automatic speech recognition (ASR) and text-to-speech (TTS). These systems work in conjunction with natural language processing (NLP) to analyze the user's intent and formulate a response, thereby completing the structure of a conversational AI agent.

ASR and TTS are two distinct fields in speech processing research. In recent years, ASR has made significant strides through self-supervised pre-training [3]–[7]. These methods can learn meaningful representations of speech even without labels and are therefore readily applicable to unlabelled or low-resource settings. Another approach, using large amounts of internet audio to generalize to any dataset without fine-tuning, was explored in Whisper [7], which is also a multi-task model. TTS is often divided into two stages: feature generation and waveform synthesis. Feature generation is where the text is transformed to intermediate features, such as spectrograms, and waveform synthesis is where the intermediate features are finally transformed into audio. Recent approaches take advantage of attention-based [8]–[10] or flow-based architectures [11], [12] to generate mel spectrograms. Vocoders based on Generative Adversarial Networks (GAN) are then used for the final audio synthesis [13]–[16]. While ASR and TTS are often developed independently, there are studies exploring simultaneous and closed-loop setup training of both [17]–[20].

Since speech and text are almost always used together in a system, it seems intuitive to explore a line of work that focuses on leveraging knowledge from the two modalities. The acoustics from speech and the semantics from text can be combined by developing a joint embedding space for both modalities [21]. These joint embedding spaces are used in various cases such as speech translation (ST), semantic

matching, and spoken language understanding (SLU) [22]–[26].

The remainder of this paper is organized as follows: Section II discusses the recent advances in joint speech-text embeddings and ASR-TTS joint training, Sections III and IV describe the full model architecture and experimental setup, Section V provides the results and analysis in each tasks, and finally, Section VI presents the conclusions and possible future directions to extend this research.

II. REVIEW OF RELATED LITERATURE

The two main concepts in this study are joint speech-text embedding space and ASR-TTS joint training.

The main purpose of unifying the embedding space between speech and text is to take advantage of the knowledge contained in each modality. In Chung et al.'s work [22], an unsupervised framework to align speech and text embedding spaces through adversarial training was proposed. They compared the proposed unsupervised framework to its supervised alignment counterpart and achieved comparable results in spoken word classification and translation. An SLU model based on a combination of pretrained ASR and natural language understanding (NLU) was presented by Denisov et al. [23]. A teacher-student framework was used to minimize the distance between speech and text features. XST-Net [25] uses modality and language indicators for speech and text inputs before feeding the embeddings to a shared transformer encoder for speech-to-text translation task. FAT-MLM [26] is a single encoder-decoder framework which accepts speech and text inputs through concatenation. Another teacher-student framework was presented in Huzaifah et al.'s work [21] which uses a pre-trained language model as the defined space. SLAM [27] uses initial and shared conformer encoders for speech and text modalities on self-supervised learning objectives for text (BERT [28]) and speech (w2v-BERT [29]). MAESTRO [30] uses a self-supervised training method, sequence alignment, duration prediction, and embedding-matching using aligned masked-language model loss, to unify the speech and text representation space for ASR and ST tasks. Virtuoso [31] extends this framework for the TTS task. In Gonzales et al.'s work [32], pre-trained models are used for initial embedding extraction before modality matching, and a mutual information estimator to explicitly disentangle speaker and non-speaker information.

Joint training of ASR and TTS is mainly influenced by the speech chain mechanism of human communication [33]. An earlier work by Karita et al. [18] used ASR, TTS, and autoencoders with inter-domain loss between speech and text for semi-supervised multitask training. Another similar approach is explored by Ren et al. [19] where denoising autoencoders are used without explicit need for speech and text feature matching. A fully closed-loop speech chain mechanism was developed in Tjandra et al.'s work [17] leveraging both paired and unpaired speech and text data. In Hori et al.'s work [34], the cycle-consistency method was proposed to improve the ASR performance while utilizing a text-to-speech-similar setup for reconstruction. A work by Makishima et al. [20] improves upon the cycle-consistency-based training to develop not only the ASR model but also for the TTS performance. A more recent work was presented in VoxtLM [35] where a unified decoder-only language model was used to integrate text and speech for four tasks: speech recognition, speech synthesis, text generation, and speech continuation.

This paper presents a proof-of-concept architecture that takes advantage of the joint speech-text embedding space for three speech processing tasks: speaker recognition; ASR; and TTS. This investigates the feasibility of simultaneous training using a shared encoder for the speech and text modalities, and distinct decoders for each task. While other prior works only aim to bring speech and text closer in the feature space through alignment or distance minimization [21]–[23], the proposed architecture follows the shared encoder setup [27], [30]–[32] because the goal is to have the same feature representation regardless of the input modality. For the joint training, the architecture employs a simple multi-task setting to avoid too much interference or dependency between the tasks. The main motivation for developing this is to minimize the memory footprint required to run each task, thus making it a viable option for deployment to low-resource and memory-constrained devices. Additionally, the setup of the architecture yields information-rich features which can be readily used for other tasks, such as voice conversion. Sample outputs are available in a GitHub repository[1] while the code and checkpoints will be released upon the publication of the article.

III. MODEL ARCHITECTURE

The voice-based conversational AI found in devices such as smart speakers commonly consist of ASR for the speech input, NLP for the language analysis and response formulation, and TTS for the speech output. This model only focuses on the ASR and TTS, with an addition of speaker recognition.

The entire model is composed of a speech encoder, text encoder, alignment encoder, duration predictor, shared encoder, speaker recognition, token decoder, and mel spectrogram decoder. Figures 1 and 2 show the model and data flow during training and inference, and how each module is connected.

A. SPEECH ENCODER

The speech encoder serves as the initial feature extraction for speech input. This is based on the Wav2Vec2.0 [4] feature encoder. It consists of several 1D-convolutional blocks followed by a GELU activation function. The purpose of this block is to extract a latent speech representation from the raw input waveform and to downsample the signal to a lower resolution. The outputs are similar to a mel spectrogram in function, but the features are learned through training and expected to contain both speaker and non-speaker information. This module is needed whenever a speech input is required

[1] https://github.com/C3Imaging/joint-speech-text

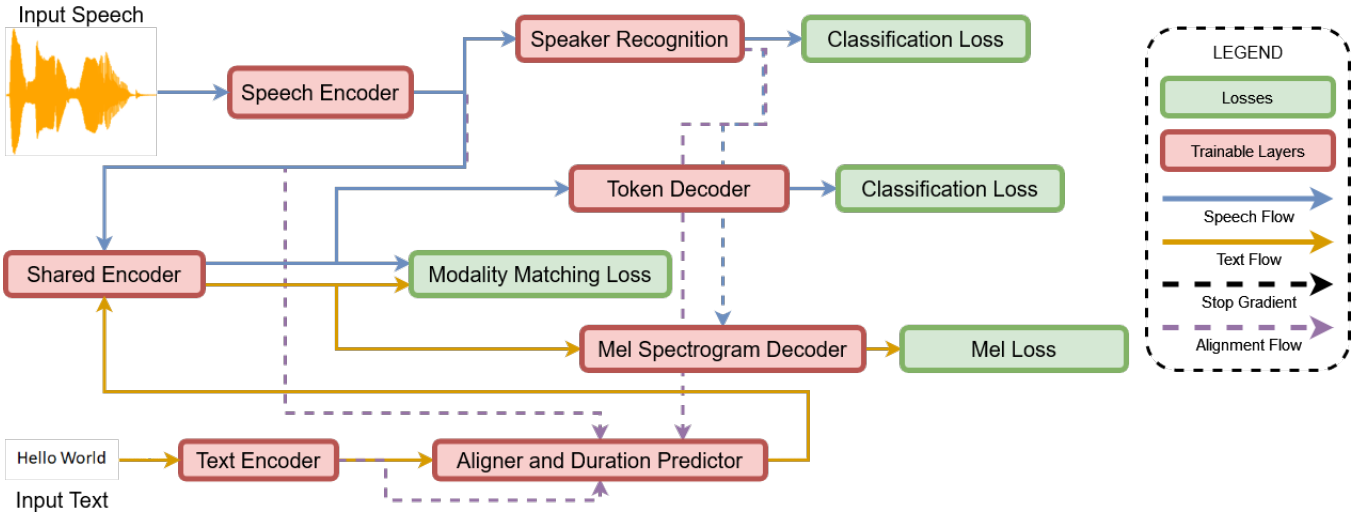

FIGURE 1. **Model Architecture: Training Setup. Paired speech and text inputs enter the model through their respective modality encoders. The encoded speech input goes through the speaker recognition module for the first task. The same encoded speech also goes through the shared encoder and to the token decoder for the ASR task. On the other hand, the encoded text are first aligned and resampled to match the encoded speech's duration, before entering the shared encoder. Afterwards, it enters the mel spectrogram decoder for the TTS task. The shared encoder outputs of the encoded speech and text are matched through the modality matching loss.**

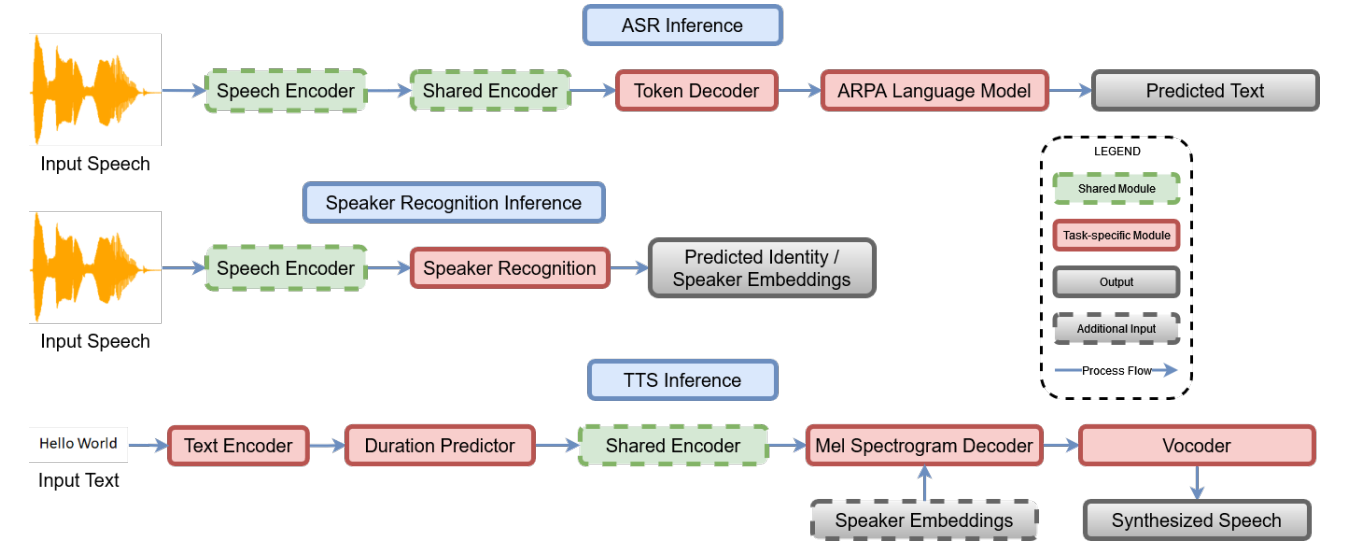

FIGURE 2. **Model Architecture: Inference Setup. The modules being used in the system are dependent on which task. For the ASR, the input speech goes through the same modules as in the training setup with an additional language model to output the predicted text. For the speaker recognition, no additional modules are concatenated to produce the speaker embedding and the predicted identity. For the TTS task, a vocoder is used to synthesize the mel spectrograms to a speech waveform.**

for a specific task, i.e. ASR, speaker recognition, and speaker embedding extraction.

B. TEXT ENCODER

Like the speech encoder, the text encoder serves as the initial feature extraction for text input. This is a simple trainable embedding layer followed by conformer encoders to extract text-specific representations. Conformer encoders [36] consist of two feed forward layers with half-step residual connection, sandwiching a multi-headed self-attention layer and a 1D-convolution layer, followed by a layernorm. These models are capable of capturing local and global dependencies of a sequence through convolution and attention heads.

C. ALIGNMENT ENCODER AND DURATION PREDICTOR

The speech and text features will have different lengths after the initial encoders. As stated in MAESTRO [30], merging these two without alignment will result in poor performance. The purpose of these modules is to align and upsample the text features to match the length of the speech features. The aligner module is based on Badlani et al.'s work [37]. It is an unsupervised method using forward-sum, Viterbi, and a static prior, to learn the alignments and durations between text and speech. These durations are subsequently used by the duration

predictor which is based on FastSpeech [38]. Additionally, the text features are first conditioned on speaker embeddings before entering the duration predictor. The aligner module is trained using forward-sum and binarization losses, while the duration predictor is trained using mean-squared error loss in the log domain.

As seen in Figure 1, stop-gradients are employed in these modules during training. The gradients computed from the losses in these two modules are not propagated to the rest of the system and losses from other modules do not affect these.

D. SHARED ENCODER

The shared encoder accepts both speech and text features as inputs. This is composed of several conformer encoder blocks similar to the text encoder. During training, paired speech-text inputs are fed into the module and the mean absolute error (MAE/L1) loss is calculated between the outputs as the modality matching loss. This forces the features to be identical regardless of the input modality, compared to using a distance metric which only brings the features closer in the embedding space. Theoretically, the output of the shared encoder contains speaker-independent information.

E. SPEAKER RECOGNITION

The speaker recognition module is based on the Multi-scale Feature Aggregation (MFA) Conformer [39]. The reference work uses 80-dimensional filter banks, while this configuration uses learned speech representation. The final layer outputs before the classification layer are the speaker embeddings which are also used by the mel spectrogram decoder. The module is trained using Additive Margin Softmax (AMSoftmax) [40] loss.

F. TOKEN DECODER

The token decoder module is composed of a single feed forward layer, which projects the shared encoder outputs to the vocabulary size, making it the classification layer. The module is trained using the Connectionist Temporal Classification (CTC) loss.

G. MEL SPECTROGRAM DECODER

The mel spectrogram decoder module is also composed of Conformer blocks to decode mel spectrograms and 1D-convolutional blocks that act as the postnet, similar to Tacotron 2 [41]. Before entering the decoder, the speaker embedding output of the speaker recognition module is projected and added to the shared encoder outputs. The module is trained using MAE/L1 loss that are computed before and after the postnet.

H. SHARED LAYERS AND LOSSES

In summary, there are four losses that propagate in the entire system: speaker recognition, modality matching, ASR, and TTS losses. Based on Figure 2, the speech and shared encoders are used on more than one task. The speech encoder is trained on the losses of the three tasks: speaker recognition; ASR; and modality matching, while the shared encoder is trained on all but the speaker recognition. Since no task is more important than the others, no loss weighing is employed for all the experiments.

IV. EXPERIMENT SETUP

Standard metrics were used for each task to evaluate overall performance. Word and character error rates (WER/CER) were computed for the ASR task. A pretrained statistical language model was used through shallow fusion to augment the ASR performance. It is the 4-gram ARPA LM trained on LibriSpeech dataset[2]. For the speaker recognition, predicted identity accuracy were used for the training speakers and a t-distributed Stochastic Neighbor Embedded (t-SNE) plot for the speakers not seen during training. For the TTS task, a vocoder based on Multi-Band MelGAN [15] and customized parameters was trained on the training set ground-truth mel spectrograms to synthesize the audio waveforms[3]. Mel Cepstral Distortion (MCD)[4] was calculated between the synthesized and ground-truth speech waveforms. Dynamic Time Warping (DTW) was used when computing MCD to accommodate temporal differences. Two Mean Opinion Score prediction networks, NISQA [42] and SSL-MOS [43], were also used to relatively compare the synthesis outputs. These prediction networks can scale easily compared to subjective evaluations and while these prediction systems are not absolute reflections of subjective MOS, they are enough to provide insights for the relative performance of the models.

The dataset used for the experiments was the CSTR VCTK Corpus [44]. It contains a total of 109 speakers with about 400 sentences each, averaging 20 minutes per speaker. 9 speakers were randomly chosen for the "unseen" test set and the remaining 100 speakers have 5% of their total number of utterances randomly chosen for the "seen" test set. In total, there were 43 male and 57 female speakers in the seen test set, and 3 male, 6 female speakers in the unseen test set. Moreover, the voice conversion evaluation set incorporated 670 paired permutations of same utterances but with different speakers, from the seen test set.

Min-max normalization was performed first on the raw speech waveform before entering the model. 80-dimensional mel spectrograms were extracted using a hop size of 320 samples and a window length of 1280 samples. This is different to the usual mel spectrogram extraction parameters, and the reason for that is to match the output resolution of the speech encoder while also retaining the window length-hop size ratio. The text inputs/targets were converted into phonemes using g2pE [45] for the text encoder input, but retained the uppercase letter vocabulary for the ASR targets.

The speech encoder followed the setup of Wav2Vec2.0 [4] with seven 1D-convolution layers of kernel sizes [10, 3,

[2]http://openslr.org/11
[3]https://github.com/kan-bayashi/ParallelWaveGAN
[4]https://pypi.org/project/pymcd

3, 3, 3, 2, 2] and strides [5, 2, 2, 2, 2, 2, 2]. The encoder took in 16 KHz raw waveform and downsampled it to a 50 Hz, 512-dimensional latent representation resolution. This module used the Hugging Face Transformers implementation of Wav2Vec2[5]. The speaker recognition module followed the MFA-Conformer setup [39], particularly the 1/2 subsampling rate. The number of conformer blocks was reduced to 3, the AMSoftmax margin was 0.4 and the scaling factor was 30. This module used the official implementation released by the authors[6].

The alignment encoder used two 1D-convolutional layers for the input text and 3 layers for the speech features. The convolutions had kernel sizes 3 for the first few layers and 1 for the last convolution layer, while all strides were 1. For the duration predictor, 2 layers of 1D-convolution, ReLU activation, LayerNorm, and Dropout were implemented. The text encoder embedding layer had a vocabulary size of 79, for the phonemes and special characters, and an embedding dimension of 384. This was followed by 4 blocks of conformer encoders with 384 attention dimensions, 2 attention heads, 1536-dimensional feed forward layers, and convolutional kernel of 7. The final output of the text encoder had 512 dimensions. The shared encoder consisted of 8 conformer blocks with 512 attention dimension, 8 attention heads, 1024-dimensional feed forward layers, and convolutional kernel of 31. The ASR module was a single feed forward network with input dimension of 512 (output dimension of the shared encoder) and output dimension of 32, which was also the vocabulary size. The TTS module was another set of 4 conformer blocks with the same parameters as the text encoder, except the convolutional kernel of 31 and output dimension of 80, which was the mel spectrogram. Finally, the postnet was composed of five 1D-convolutional layers with kernel size of 5 and channel size of 256. All modules described in this paragraph were implemented using the ESPNet toolkit [46].

All modules were trained simultaneously with a static learning rate of 1e-4. No loss weighing was applied and there was no early stop or layer freezing for any modules throughout the training process. The model was trained on 2 Nvidia Titan RTX, with a batch size of 2, and gradient accumulation was applied for 4 steps, making an effective batch size of 16 per optimization steps. The whole architecture was trained for 450 epochs.

Additionally, two single-task equivalent models of the joint architecture were trained for comparison: ASR-only and TTS-SpkRecog. The purpose of these models is to provide a baseline performance between training a single-task model and the proposed multitask architecture. These models followed the same specifications of the joint model but the modules present are only those relevant to the task itself. The ASR-only was trained for 175 epochs, while the TTS-SpkRecog was trained for 130 epochs. Figure 3 shows the diagram of the detached models and the associated modules for each and Table 11 in Appendix A provides a detailed breakdown of which modules are present for these baseline models.

[5]https://huggingface.co/docs/transformers/en/index
[6]https://github.com/zyzisyz/mfa_conformer

V. RESULTS

A. MEMORY FOOTPRINT

Table 1 shows the model sizes for each setup. By sharing the encoder layer, there was minimal increase in the number of parameters and memory size for the joint architecture. In summary, there is 28.17% decrease in memory footprint for using the Joint model. For fairer comparisons between the detached and joint performance, the setups were retained where possible. The module breakdown is also available in Appendix A. In the same table, details about the state-of-the-art (SotA) models for ASR and TTS are also included.

B. SPEAKER RECOGNITION

The speaker recognition module obtained an accuracy of 97.64% accuracy in the test set. To visualize the 256-dimensional speaker embeddings, t-SNE technique was applied and results are shown in Figure 4. The top plot shows 10 random speakers clearly clustered while the bottom plot shows the 10 unseen speakers with minimal clustering. The module learned the training speakers' characteristics as evidenced by the high accuracy and clustering, but overfitted to them as shown in the unseen speaker plot. This can be remedied by increasing the number of speakers in the training set, hence increasing the model's generalizability capabilities.

To further contextualize the results, Table 2 shows reported accuracies in literature using the VCTK dataset. It is worth noting that even if the same dataset were used, the training and testing splits, as well as experimental conditions, were different for each. All reported results, including the proposed model, are above 95%.

C. AUTOMATIC SPEECH RECOGNITION

For the ASR module, Word and Character Error Rates (WER/CER) were calculated using the torchmetrics python package[7]. To assess whether the model can actually predict the right tokens, metrics were computed with and without the help of the ARPA LM. A summary of the calculated metrics are shown in Table 3. The seen and unseen speaker test set were combined and used for the evaluation.

Without a language model, the Joint architecture has a 23.49% WER and a 9.10% CER compared to the 15.56% and 5.60% WER and CER of the ASR-only model. Adding the language model relatively improves the performance of the Joint model by 22.61% and 12.64% in WER and CER, but still has higher error rates compared to the ASR-only model. While both the Joint and ASR-only models are capable of transcribing meaningful words, the above 10% WER suggests that further improvements should be made. Sample outputs of the ASR task can be seen in Table 4.

[7]https://pypi.org/project/torchmetrics/

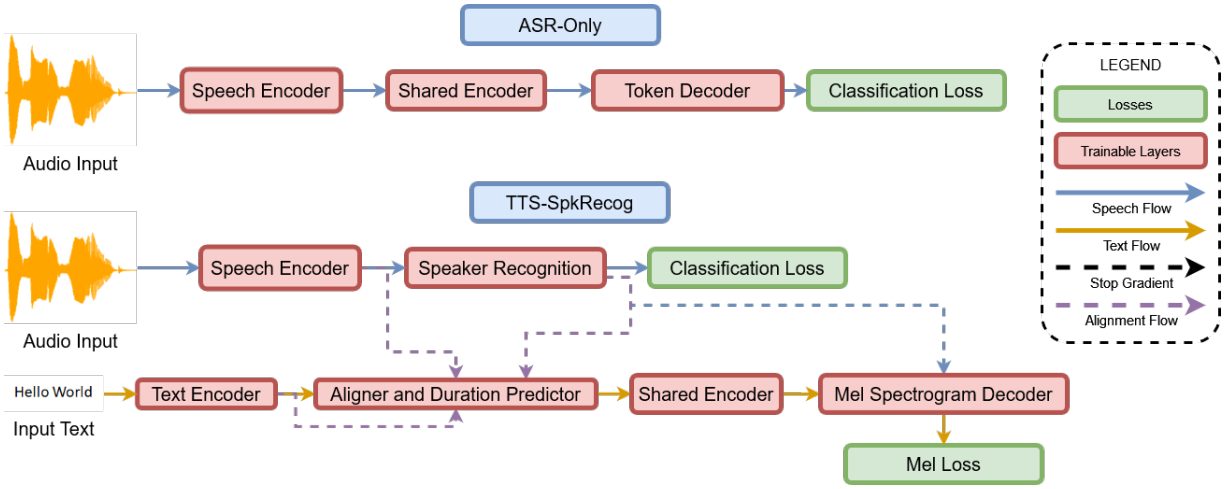

FIGURE 3. **Detached Model Setup. Each main task contained the essential modules and were not connected with the other tasks during training.**

Model	Task	Training Set	Number of Parameters	Memory Size
Wav2Vec2 [4]	ASR	LibriSpeech 960h	94.4 M	360.09 MB
Data2Vec [47]	ASR	LibriSpeech 960h	93.19 M	355.49 MB
Whisper [7]	Multilingual ASR, Speech Translation, Spoken Language Identification, Voice Activity Detection	680,000 hours of internet audio	72.59 M	276.92 MB
VoxtLM [35]	ASR, TTS, Speech Continuation, Text Generation	LibriLight, LibriSpeech, LibriTTS	315.7 M	1204.29 MB
Tacotron 2 [41]	TTS	VCTK	26.92 M	103.04 MB
FastSpeech 2 [9]	TTS	VCTK	70.46 M	269.13 MB
ASR-only	ASR	VCTK	38.28 M	146.02 MB
TTS-SpkRecog	TTS	VCTK	97.56 M	372.17 MB
Joint ASR-TTS	ASR, TTS	VCTK	97.57 M	372.23 MB

TABLE 1. **Model Size Comparison. Sizes are not additive for the experimental setup since there are layers shared in the joint architecture. This shows that three tasks can be done in one model with minimal increase in parameters and memory size.**

Literature	Technique	Accuracy
Medikonda et al. [48]	Twofold Information Set Features	98.90%
Wang et al. [49]	Prototypical Network Loss	95.63%
Chauhan et al. [50]	18 Features Aggregation	100.00%
Proposed Model	Learned Latent Features	97.64%

TABLE 2. **Speaker Recognition Accuracy on VCTK Dataset. Results from literature show above 95% speaker recognition accuracy which the proposed model is on par**

Metric	ASR-only		Joint	
	w/o LM	w/ LM	w/o LM	w/ LM
Word Error Rate (WER)	15.56%	10.77%	23.49%	18.18%
Character Error Rate (CER)	5.60%	4.56%	9.10%	7.95%

TABLE 3. **ASR Results. Word and Character Error Rates are shown for both ASR-only and the Joint model. Results with and without the language model are also shown. The Joint model has increased error rates compared to the ASR-only model.**

Table 5 shows four pretrained SotA ASR models: Wav2Vec2[8]; Data2Vec[9]; Whisper[10]; and VoxtLM[11]. These models were evaluated on the test set and are shown on the table with the Joint and ASR-only results. Results are better in the SotA models, which is expected because of the large datasets, architecture, and training complexity of such models. It is due to these reasons that a direct comparison of the proposed model and SotA models would be biased. The table is only given as a reference for the evaluation results.

[8]https://huggingface.co/facebook/wav2vec2-base-960h
[9]https://huggingface.co/facebook/data2vec-audio-base-960h
[10]https://huggingface.co/openai/whisper-base
[11]https://huggingface.co/soumi-maiti/voxtlm-k1000

Transcribed Output	Ground Truth	WER	CER
A FORMAL ANNOUNCEMENT IS EXPECTED THIS MORNING AT A NEWS CONFERENCE	A FORMAL ANNOUNCEMENT IS EXPECTED THIS MORNING AT A NEWS CONFERENCE	0.00%	0.00%
THERE WILL BE NO *SHOROTAGE* OF QUALITY APPLICANTS	THERE WILL BE NO *SHORTAGE* OF QUALITY APPLICANTS	12.50%	2.13%
SEVERALHER PUPILS AND STAFF WERE SERIOUSLY *ENJOYED* IN *A* ACCIDENT	*SEVERAL OTHER* PUPILS AND STAFF WERE SERIOUSLY *INJURED* IN *THE* ACCIDENT	36.36%	13.04%
THERE WOULD APPEAR TO BE NO MOTIVE FOR THE *TACK*	THERE WOULD APPEAR TO BE NO MOTIVE FOR THE *ATTACK*	10.00%	4.08%
HOWEVER HE ADDED *THE WORE* SIGNS OF *PROGRESSE*	HOWEVER HE ADDED *THERE WERE* SIGNS OF *PROGRESS*	37.50%	8.89%

TABLE 4. ASR Sample Outputs. Language model was used to produce the transcribed outputs. The incorrectly transcribed words are italicized in both columns and error rates were computed for each sample utterance.

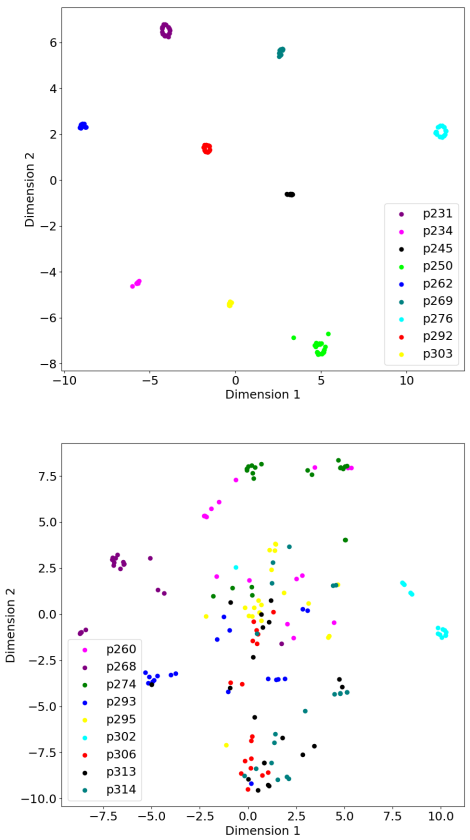

FIGURE 4. T-distributed Stochastic Neighbor Embedding (t-SNE) plots for Speaker Embeddings. Top: 10 random speaker from the seen set. Bottom: 9 speakers from the unseen set. Seen speakers are clearly clustered together compared to the unseen speakers.

Model	Word Error Rate	Character Error Rate
Wav2Vec2	8.59%	2.98%
Data2Vec	8.96%	3.30%
Whisper	7.02%	5.11%
VoxtLM	14.50%	7.67%
ASR-only	10.77%	4.56%
Joint	18.18%	7.95%

TABLE 5. State-of-the-Art (SotA) Models ASR Results. Word and character error rates on the test set using SotA pretrained models. Expectedly, results are better than the proposed model due to architecture complexity and training setup.

SEEN SPEAKERS TEST SET			
Metric	**Ground Truth**	**TTS-SpkRecog**	**Joint**
MCD	NA	4.23	4.31
NISQA	2.90	3.00	2.98
SSL-MOS	3.36	3.43	3.28
UNSEEN SPEAKERS TEST SET			
Metric	**Ground Truth**	**TTS-SpkRecog**	**Joint**
MCD	NA	4.68	4.79
NISQA	2.93	3.10	2.98
SSL-MOS	3.36	3.57	3.40

TABLE 6. TTS Results. TTS-SpkRecog scores are better than Joint model scores but only by a small margin. MCD scores are better on both models in the seen set, but NISQA and SSL-MOS are better on the unseen set.

Besides, the main goal of the proposed model is to be able to perform multiple tasks and not to establish state-of-the-art results.

D. TEXT-TO-SPEECH

Figure 5 shows three sample pairs of synthesis and target mel spectrograms. In terms of MCD, the lower the value, the better, with 0 being a perfect match between two waveforms. NISQA and SSL-MOS are trained on datasets based on the five-point absolute category for naturalness, with 5 being Excellent and 1 for Bad. To reiterate, the MOS prediction networks were only used for evaluating the relative performance of the models.

In the seen set, there is only 0.08 difference between the average MCD of the TTS-SpkRecog and Joint models, 0.02 difference for the NISQA, and 0.15 difference for the SSL-MOS. In the unseen set, the differences are slightly higher

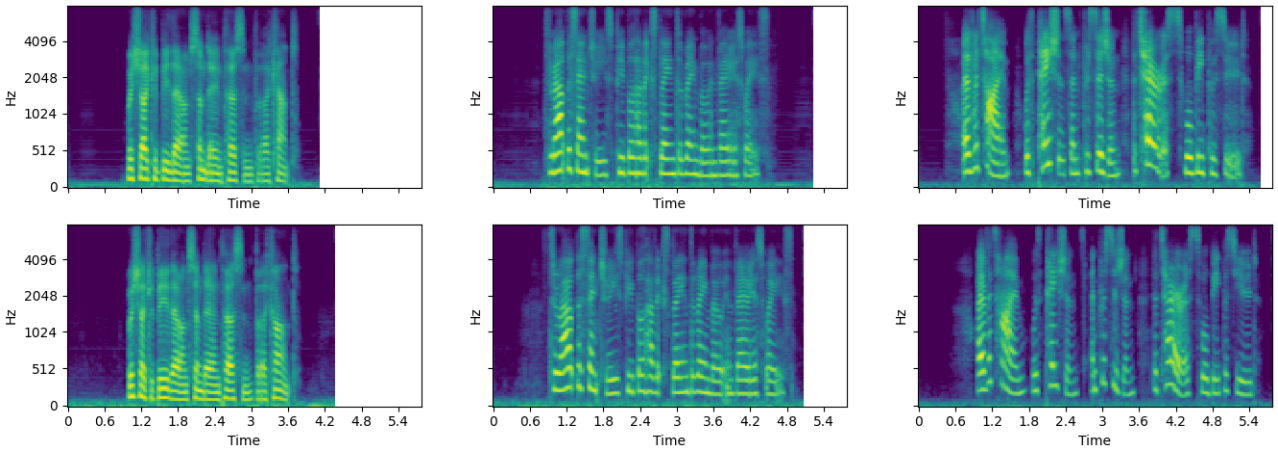

FIGURE 5. TTS Mel Spectrograms. First row: synthesized samples, Second row: target outputs, each column are different pairs. Each synthesized sample resembles their target counterparts and there are minimal duration differences.

Model	MCD	NISQA	SSL-MOS
Ground Truth	-	2.90	3.36
Tacotron 2	7.74	2.38	2.35
FastSpeech 2	6.58	2.89	2.84
TTS-SpkRecog	4.25	3.00	3.44
Joint	4.33	2.97	3.29

TABLE 7. State-of-the-Art (SotA) Models TTS Results. Tacotron 2 and FastSpeech 2 models were evaluated on the same combined seen and unseen test set. Scores for MCD, NISQA, and SSL-MOS were computed for all models.

Metric	Ground Truth	Joint
MCD	NA	5.22
NISQA	2.90	2.98
SSL-MOS	3.32	2.73

TABLE 8. VC Results. MCD score is higher, NISQA are the same, and SSL-MOS are lower, when compared to the TTS task.

with 0.11, 0.12, and 0.17 for MCD, NISQA, and SSL-MOS, respectively. Based on the samples in Figure 5, the Joint model can decode mel spectrograms which synthesizes to fairly natural speech and the intended identity. While all metrics are better for the TTS-SpkRecog model, numbers show that there is minimal performance drop in the TTS task of the Joint model. The scores for the unseen set are also good indications that the TTS module can still synthesize speech even on unseen speakers.

Table 7 shows two pretrained SotA TTS models: Tacotron 2[12] and FastSpeech 2 [13] which were used together with a Parallel WaveGAN [13] vocoder[14]. These models were also trained in the VCTK dataset (but with different train-test splits) and were evaluated on the same test set used in the proposed architecture. For all scores, the proposed architecture shows better performance than the SotA models.

E. VOICE CONVERSION

Referring to Figure 2, if the input is audio/speech, it will go through the speech encoder and shared encoder to output the encoded acoustic embeddings. However, instead of using it for the token decoder, by inputting it to the mel spectrogram decoder, together with another speech input for the target speaker embeddings, the resulting process is voice conversion. This is only possible since the model forces speech and text inputs to have the same encoded features through the modality matching loss during training.

Table 8 summarizes the results for the VC task. While the mean MCD is higher and the SSL-MOS is lower than the TTS results, the NISQA is the same. Figure 6 shows two conversion samples, one male-to-female and one female-to-male. Visually, the synthesized mel spectrograms are close to that of the source mel spectrograms, but the model is capable of making fine changes to transform the identity of the speaker, using a random sample of the target. Additionally, the duration of speech is not changed between the source and synthesized output.

F. CONFLICTING GRADIENTS

Multitask learning has an innate problem of interference between different tasks which sometimes lead to reduced overall performance. When comparing the Joint model to the detached models, the most significant performance drop can be found in the ASR task. Looking back at the model architecture in Figure 2, the ASR branch has the two shared modules, speech and shared encoder, and therefore experiences the most interference. To quantify this interference, the amount of conflicting gradients were computed in both shared modules.

[12]https://huggingface.co/espnet/kan-bayashi_vctk_xvector_tacotron2
[13]https://huggingface.co/espnet/kan-bayashi_vctk_xvector_conformer_fastspeech2
[14]https://drive.google.com/open?id=1qoocM-VQZpjbv5B-zVJpdraazGcPL0So

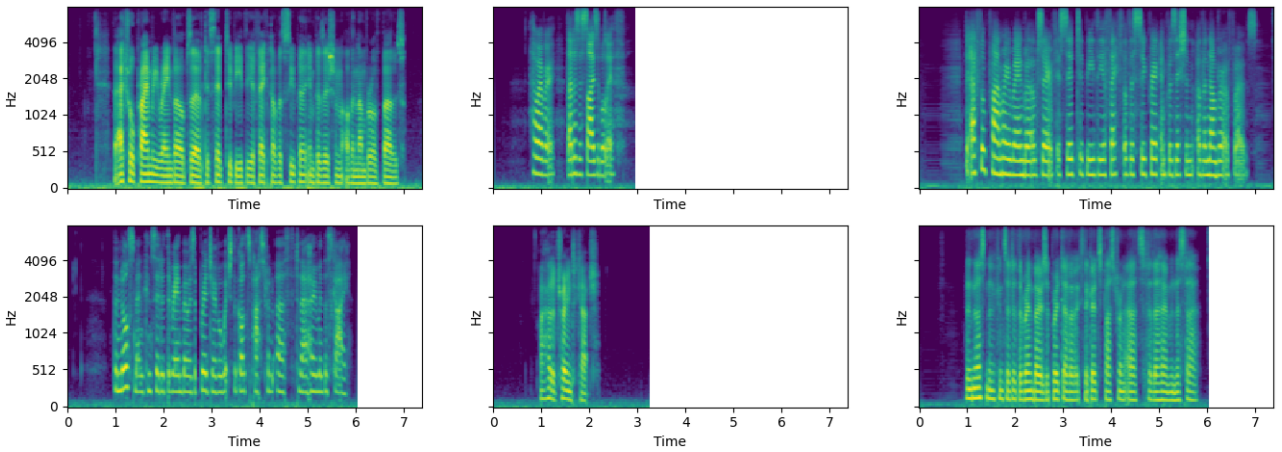

FIGURE 6. VC Mel Spectrograms. First column: source samples, Second column: reference speakers, Third column: synthesized outputs, each row is a different source-reference-synthesis triplets. First row is male-to-female while second row is female-to-male conversion.

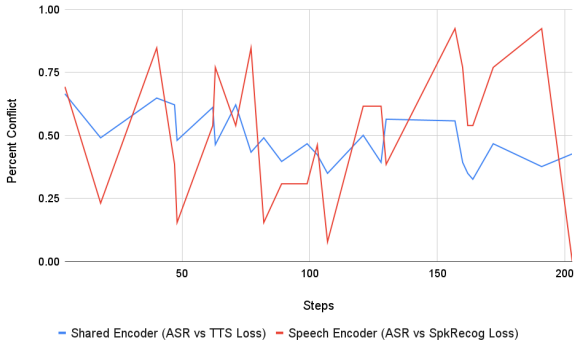

FIGURE 7. Conflicting Gradients. Blue: Shared Encoder, Red: Speech Encoder. The plot shows the percentage amount of conflicting gradients in each module for the first 200 steps of epoch 451.

Statistic	Shared Encoder	Speech Encoder
Average	49.14%	48.76%
Standard Deviation	9.83%	25.50%
Minimum	22.82%	0%
Maximum	72.48%	100%

TABLE 9. Conflicting Gradients. The percentage amount of gradients with different directions for each step in epoch 451. Almost 50% of all layers have conflicting directions which interferes with the ASR training.

Metric	ASR-Only	TTS-SpkRecog	Joint
Number of Parameters	38.28 M	97.56 M	97.57 M
Memory Size	146.02 MB	372.17 MB	372.23 MB
Automatic Speech Recognition			
Word Error Rate	10.77%	-	18.18%
Character Error Rate	4.56%	-	7.95%
Text-to-Speech (Seen/Unseen)			
Mel Cepstral Distortion	-	4.23/4.68	4.31/4.79
NISQA	-	3.00/3.10	2.98/2.98
SSL-MOS	-	3.43/3.57	3.28/3.40

TABLE 10. Summary of Results. For easier comparison, ASR and TTS results of the Joint and detached models are shown in the table. The Joint model has significantly reduced the memory footprint of doing all three tasks compared to using individual models for each task.

Conflicting gradients, as defined in [51], are gradients of each task where the cosine similarity is less than 0. This does not include the magnitude difference of each gradient, only the direction.

Figure 7 shows the percentage of layers with conflicting gradients in each module for the first 200 steps of epoch 451. The shared encoder has 298 sets of parameters and the speech encoder has 13 sets. The ASR loss and the TTS loss gradients were computed in the shared encoder while the ASR loss and Speaker Recognition loss gradients were computed in the speech encoder. Table 9 shows the summary of the computed statistics. Overall, there are about 50% of conflicting gradients in every training step, which interfered with the ASR task, causing a lower performance compared to the ASR-only model.

For easier comparison, Table 10 summarizes the performance evaluation between the proposed Joint architecture and the detached models.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a model architecture that uses joint speech-text embeddings for three speech processing task is presented. The main advantage of using this architecture is to reduce the memory-footprint required to run ASR, Speaker Recognition, and TTS tasks. This particular setup shows a 28.17% decrease in the memory footprint compared to using two single-task models. A shared encoder was utilized to unify the speech and text embeddings, and the decoders are trained simultaneously. The proposed model was evaluated and compared to its detached/single-task counterparts. The speaker recognition

task had 97.64% accuracy on the seen speakers. A WER of 18.18% and CER of 7.95% were evaluated from the ASR module. For the TTS task, the results were closer to the single-task model with an MCD of 4.31, NISQA score of 2.98, and SSL-MOS score of 3.28 on the seen speaker set, and 4.79, 2.98, and 3.40 on the unseen set. While VC is not included in the training tasks, the proposed architecture can incorporate this process and was also evaluated with scores of 5.22, 2.98, and 2.73 in MCD, NISQA, and SSL-MOS.

Multitask models are good at reducing the overall memory footprint, but suffer from interference between each task. Therefore, incorporating gradient manipulation for lowering the effect of interference is a potential future direction of this research. Other directions are the use of a larger dataset, and the exploration of the model transferability to other domains such as child-speech and low-resource languages. Additionally, evaluating the speech-text embeddings on other downstream tasks such as semantic matching, speech translation, and spoken language understanding, with and without fine-tuning, will be explored. Finally, experiments on porting the architecture to low-resource and memory-constrained devices will be investigated.

APPENDIX A
FULL MODEL BREAKDOWN

Module	Memory	Parameters	ASR-only	TTS-SpkRecog
Speech Encoder	17.03 MB	4.46 M	YES	YES
Text Encoder	55.10 MB	14.44 M	NO	YES
Shared Encoder	128.93 MB	33.80 M	YES	YES
Aligner	11.01 MB	2.89 M	NO	YES
Duration	2.76 MB	0.72 M	NO	YES
Spk Recog	97.06 MB	25.44	NO	YES
ASR	0.06 MB	0.02 M	YES	NO
TTS	60.29 MB	15.80 M	NO	YES

TABLE 11. Full Model Breakdown. This table shows the memory size, number of parameters for each module, and also which modules are found in the detached models. The Joint model contains all modules.

Table 11 shows the memory size, number of parameters, and confirmation if the module is present in the detached models. For the ASR-only model, the text encoder, aligner, duration, speaker recognition, and TTS modules are not present. On the other hand, only the ASR module is not included in the TTS-SpkRecog model.

APPENDIX B
TRAINING RESULTS

This section gives more details about the training results of the detached and joint models. Losses and evaluation metrics per 10 checkpoints are shown for each task.

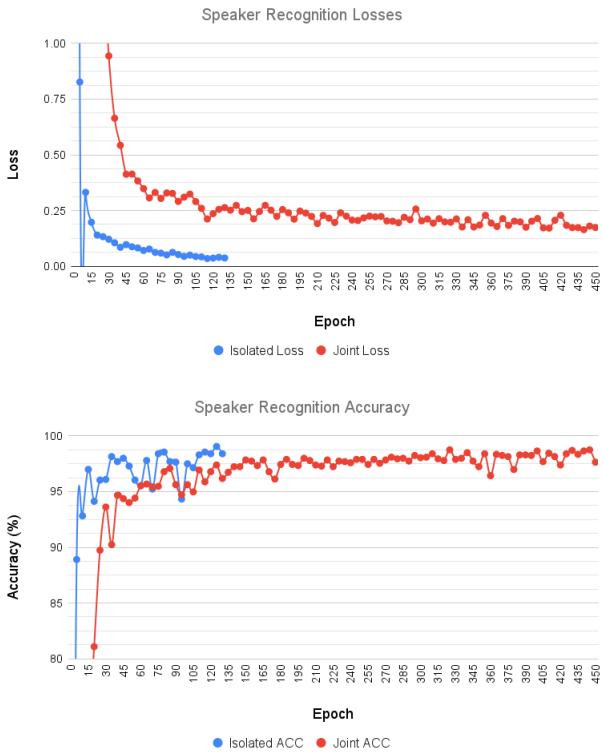

FIGURE 8. Speaker Recognition Loss and Accuracy. Blue: TTS-SpkRecog, Red: Joint. Plots show that while Joint model losses does not come close to TTS-SpkRecog, the accuracy difference is still minimal.

A. SPEAKER RECOGNITION

Figure 8 shows the training losses and seen test set accuracy of the speaker recognition task. Looking at the losses, the Joint model converged at around epoch 110 and minimal loss changes followed afterwards. However, both models already have above 95% accuracy even before epoch 100. By the end of training, the Joint model has 97.64% accuracy at epoch 450 and the TTS-SpkRecog model has 98.39% accuracy at epoch 130. Overall, there is no difference in the performance between the two models.

B. AUTOMATIC SPEECH RECOGNITION

Training losses and seen test set WER and CER plots for ASR-only and Joint model are found in Figure 9. By the end of training, the Joint model has almost the same losses as the ASR-only model. However, this does not translate to similar performance as seen in the WER and CER plots. There is about 7.78% and 3.50% difference in WER and CER by epoch 175 of ASR-only and epoch 450 of Joint. The reason for this performance difference is discussed in Section V under Conflicting Gradients.

C. TEXT-TO-SPEECH

Plots for the training losses and MCD for the TTS task, evaluated in the seen test set, are found in Figure 10. Both models have almost stable MCD by epoch 100 and minimal

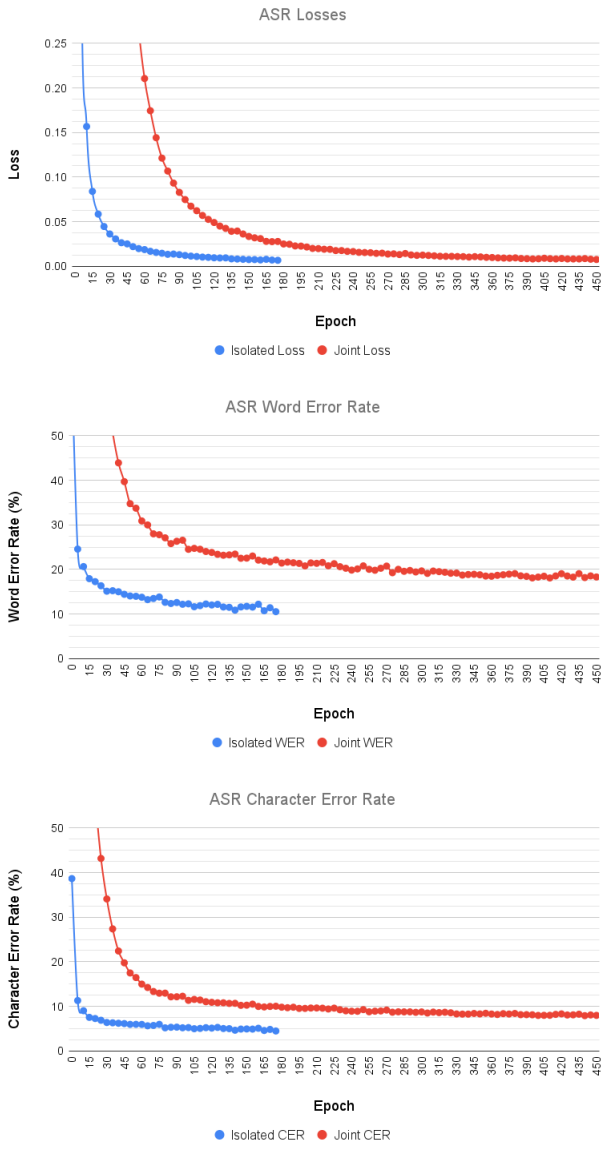

FIGURE 9. ASR Loss and Accuracy. Blue: ASR-only, Red: Joint. Losses for both models are almost equal by the end of training but the error rates are not close to each other.

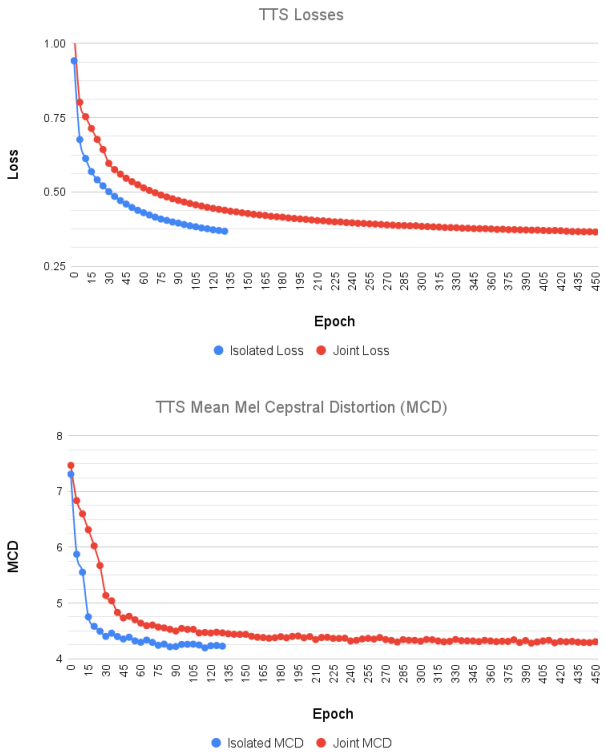

FIGURE 10. TTS Loss and MCD. Blue: TTS-SpkRecog, Red: Joint. The losses are almost equal by the end of training and the MCD values are close to each other.

changes afterwards. By the end of training there is only 0.08 MCD difference between epoch 130 of TTS-SpkRecog and epoch 450 of Joint. Overall, there is no performance difference, similar to the speaker recognition task.

REFERENCES

[1] L. Wang, N. Huang, Y. Hong, L. Liu, X. Guo, and G. Chen, "Voice-based ai in call center customer service: A natural field experiment," *Production and Operations Management*, vol. 32, no. 4, pp. 1002–1018, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/poms.13953

[2] B. L. Zhenyuan Zhang and L. Liu, "The impact of ai-based conversational agent on the firms' operational performance: Empirical evidence from a call center," *Applied Artificial Intelligence*, vol. 37, no. 1, p. 2157592, 2023. [Online]. Available: https://doi.org/10.1080/08839514.2022.2157592

[3] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.

[4] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460.

[5] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 29, p. 3451–3460, 10 2021.

[6] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[7] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 28 492–28 518. [Online]. Available: https://proceedings.mlr.press/v202/radford23a.html

[8] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 6706–6713.

[9] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=piLPYqxtWuA

[10] A. Łańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6588–6592.

[11] C. Miao, S. Liang, M. Chen, J. Ma, S. Wang, and J. Xiao, "Flow-tts: A non-autoregressive network for text to speech based on flow," in *ICASSP 2020*

- 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 7209–7213.

[12] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 8067–8077.

[13] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6199–6203.

[14] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 17 022–17 033.

[15] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen, and L. Xie, "Multi-band melgan: Faster waveform generation for high-quality text-to-speech," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 492–498.

[16] K. Song, J. Cong, X. Wang, Y. Zhang, L. Xie, N. Jiang, and H. Wu, "Robust melgan: A robust universal neural vocoder for high-fidelity tts," in *2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2022, pp. 71–75.

[17] A. Tjandra, S. Sakti, and S. Nakamura, "Machine speech chain," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 976–989, 2020.

[18] S. Karita, S. Watanabe, T. Iwata, M. Delcroix, A. Ogawa, and T. Nakatani, "Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6166–6170.

[19] Y. Ren, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Almost unsupervised text to speech and automatic speech recognition," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 5410–5419.

[20] N. Makishima, S. Suzuki, A. Ando, and R. Masumura, "Speaker consistency loss and step-wise optimization for semi-supervised joint training of TTS and ASR using unpaired text data," in *Proc. Interspeech 2022*, 2022, pp. 526–530.

[21] M. Huzaifah and I. Kukanov, "An analysis of semantically-aligned speech-text embeddings," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 747–754.

[22] Y.-A. Chung, W.-H. Weng, S. Tong, and J. Glass, "Unsupervised cross-modal alignment of speech and text embedding spaces," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[23] P. Denisov and N. T. Vu, "Pretrained Semantic Speech Embeddings for End-to-End Spoken Language Understanding via Cross-Modal Teacher-Student Learning," in *Proc. Interspeech 2020*, 2020, pp. 881–885.

[24] P.-A. Duquenne, H. Gong, and H. Schwenk, "Multimodal and multilingual embeddings for large-scale speech mining," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 15 748–15 761.

[25] R. Ye, M. Wang, and L. Li, "End-to-End Speech Translation via Cross-Modal Progressive Training," in *Proc. Interspeech 2021*, 2021, pp. 2267–2271.

[26] R. Zheng, J. Chen, M. Ma, and L. Huang, "Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation," 2021. [Online]. Available: https://arxiv.org/abs/2102.05766

[27] A. Bapna, Y. an Chung, N. Wu, A. Gulati, Y. Jia, J. H. Clark, M. Johnson, J. Riesa, A. Conneau, and Y. Zhang, "Slam: A unified encoder for speech and language modeling via speech-text joint pre-training," 2021.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[29] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "w2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 244–250.

[30] Z. Chen, Y. Zhang, A. Rosenberg, B. Ramabhadran, P. J. Moreno, A. Bapna, and H. Zen, "MAESTRO: Matched Speech Text Representations through Modality Matching," in *Proc. Interspeech 2022*, 2022, pp. 4093–4097.

[31] T. Saeki, H. Zen, Z. Chen, N. Morioka, G. Wang, Y. Zhang, A. Bapna, A. Rosenberg, and B. Ramabhadran, "Virtuoso: Massive multilingual speech-text joint semi-supervised learning for text-to-speech," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[32] M. G. Gonzales, P. Corcoran, N. Harte, and M. Schukat, "Joint speech-text embeddings with disentangled speaker features," in *2023 34th Irish Signals and Systems Conference (ISSC)*, 2023, pp. 1–5.

[33] P. B. Denes and E. N. Pinson, *The Speech Chain*. Worth Publishers, 1993.

[34] T. Hori, R. Astudillo, T. Hayashi, Y. Zhang, S. Watanabe, and J. Le Roux, "Cycle-consistency training for end-to-end speech recognition," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6271–6275.

[35] S. Maiti, Y. Peng, S. Choi, J.-W. Jung, X. Chang, and S. Watanabe, "Voxtlm: Unified decoder-only models for consolidating speech recognition, synthesis and speech, text continuation tasks," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 13 326–13 330.

[36] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," 2020.

[37] R. Badlani, A. Łańcucki, K. J. Shih, R. Valle, W. Ping, and B. Catanzaro, "One tts alignment to rule them all," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6092–6096.

[38] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[39] Y. Zhang, Z. Lv, H. Wu, S. Zhang, P. Hu, Z. Wu, H. yi Lee, and H. Meng, "MFA-Conformer: Multi-scale Feature Aggregation Conformer for Automatic Speaker Verification," in *Proc. Interspeech 2022*, 2022, pp. 306–310.

[40] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[41] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783.

[42] G. Mittag and S. Möller, "Deep Learning Based Assessment of Synthetic Speech Naturalness," in *Proc. Interspeech 2020*, 2020, pp. 1748–1752.

[43] E. Cooper, W.-C. Huang, T. Toda, and J. Yamagishi, "Generalization ability of mos prediction networks," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8442–8446.

[44] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)," 2019.

[45] K. Park and J. Kim, "g2pe," https://github.com/Kyubyong/g2p, 2019.

[46] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[47] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A general framework for self-supervised learning in speech, vision and language," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 7 2022, pp. 1298–1312.

[48] J. Medikonda, S. Bhardwaj, and H. Madasu, "An information set-based robust text-independent speaker authentication," *Soft Computing*, vol. 24, p. 5271–5287, 2020. [Online]. Available: https://doi.org/10.1007/s00500-019-04277-9

[49] J. Wang, K.-C. Wang, M. T. Law, F. Rudzicz, and M. Brudno, "Centroid-based deep metric learning for speaker recognition," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3652–3656.

[50] N. Chauhan, T. Isshiki, and D. Li, "Text-independent speaker recognition system using feature-level fusion for audio databases of various sizes," *SN Computer Science*, vol. 4, no. 531, 2023. [Online]. Available: https://doi.org/10.1007/s42979-023-02056-w

[51] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 5824–5836.

MICHAEL GIAN GONZALES (Member, IEEE) received the B.S. degree in electronics and communications engineering from the University of the Philippines Diliman, Quezon City, Philippines, in 2019. He is currently pursuing his Ph.D. degree in computer science at the University of Galway, Galway, Ireland.

From 2019 to 2022, he was a Software Developer with a startup company based in the Philippines. His research interests include machine learning and artificial intelligence in the domain of speech processing, automatic speech recognition, text-to-speech, speaker recognition, and edge device processing.

PETER CORCORAN (Fellow, IEEE) holds the Personal Chair in electronic engineering at the School of Engineering, University of Galway (formerly known as National University of Ireland Galway). He is currently an IEEE Fellow recognized for his contributions to digital camera technologies, notably in-camera redeye correction and facial detection.

He was a Co-Founder in several start-up companies, notably FotoNation, now the Imaging Division of Xperi Corporation. He has over 600 technical publications and patents, over 100 peer-reviewed journal articles, 120 international conference papers, and a co-inventor of more than 300 granted U.S. patents. He is a member of the IEEE Consumer Electronics Society for over 25 years. He is the Editor-in-Chief and the Founding Editor of IEEE Consumer Electronics Magazine.

NAOMI HARTE (Member, IEEE) received the B.A.I degree in electronic engineering from Trinity College Dublin, in 1995, and the Ph.D. degree in engineering from Queen's University Belfast, in 1999.

From 1999 to 2008, she worked with various start-ups in the field of DSP Systems Development, including her own company. She is currently a Professor in Speech Technology with the School of Engineering in Trinity College Dublin. She is also a Co-PI and founding member of ADAPT SFI Centre where she leads research on multimodal interaction. Her research interests include audio-visual speech recognition, speech synthesis evaluation, multimodal speech analysis

Prof. Harte earned the Google Faculty Award in 2018 and was shortlisted for the AI Ireland Awards in 2019.

MICHAEL SCHUKAT (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science and medical informatics from the University of Hildesheim, Germany, in 1994 and 2000, respectively. He is currently an associate professor with the School of Computer Science, University of Galway.

From 1994 to 2002, he worked in various industry positions, where he specialized in deeply embedded real-time systems across diverse domains, such as industrial control, medical devices, automotive, and network storage. His research interests include AI and its application in computer vision, cybersecurity, health informatics, and energy management.