IEEE Open Journal of Signal Processing

Received 6 September 2023; revised 2 December 2023; accepted 4 December 2023. Date of publication 14 December 2023; date of current version 29 December 2023. The review of this article was arranged by Associate Editor H. Kamper.

Digital Object Identifier 10.1109/OJSP.2023.3343342

Streaming ASR Encoder for Whisper-to-Speech Online Voice Conversion

ANASTASIA AVDEEVA, ALEKSEI GUSEV, TSEREN ANDZHUKAEV, AND ARTEM IVANOV

FluentaAI, Wilmington, DE 19803 USA

CORRESPONDING AUTHOR: ANASTASIA AVDEEVA (email: anastasia.avdeeva@fluenta.ai).

ABSTRACT Whispered speech is a quiet voice without vocalization. One of the common cases of using whispered speech is a technique that can help overcome stuttering. But whispered speech can be uncomfortable and difficult to understand in everyday communication. To address these problems, we propose a method of low-delayed whisper-to-speech voice conversion, which can be useful in real life communication of people with disordered speech. As part of our research, we study the impact of streaming Automatic Speech Recognition models on the quality of voice conversion, comparing different streaming models and methods for model adaptation to streaming settings, and showing the importance of using such models in cases of low-delayed voice conversion.

INDEX TERMS Speech recognition, voice conversion, disordered speech, whisper-to-speech processing.

I. INTRODUCTION

Despite the huge progress in developing speech processing tools for various types of disordered speech there is still room for improvement. In this research we concentrate on the stuttering problem. Based on our literature review there are only several works regarding the stuttering problem. These studies discover different aspects such as detecting stuttering type [1], recognizing and even synthesizing [2] stuttering speech. But in this investigation the focus is on a solution which can partially help to control stuttering. According to [3], one of the techniques which can help to overcome stuttering is whispered speech. But whispered speech lacks naturalness due to absence of the fundamental frequency (F0). Thus, we aim to create a system capable of transforming whispers into regular speech and apply this method to real-time processing.

The majority of novel voice conversion (VC) systems adopt the following scheme. The whole system usually consists of three parts: an Automatic Speech Recognition (ASR) encoder for phonetic posteriorgrams (PPGs) extraction, a decoder taking PPGs features as input to predict mel spectrograms of target audio and a vocoder for synthesizing audio. The acoustic-phonetic distinctions between whispered and regular speech lead to substantial degradation of ASR systems [4]. However, according to [5] a small set of whispered or pseudo-whispered data used for adaptation brings significant improvements in ASR systems quality. Thus, the model trained on a large amount of speech can be easily adapted to the whispered domain. Also, the recent breakthrough in self-supervised learning (SSL) allows to obtain well-performing ASR models having only a few hours of labeled data [6]. But, unfortunately, the design of SSL training makes streaming mode challenging for such models.

This paper proposes the following contributions:

- We demonstrate the ability of HuBERT [7] model pretrained with SSL to work in a streaming mode after an attention context masking or chunk-wise fine-tune training procedure.
- We show the importance of using a streaming encoder model to improve the quality of low latency whisper-to-speech VC.
- We propose an online VC system adapted to the whispered speech domain and evaluate the results on various test datasets.

Samples from our conversion systems and several publicly available systems can be found on the demo page.[1]

II. RELATED WORKS

A large variety of VC methods have been studied recently. However, such methods show poor audio quality when applied to whisper-to-speech VC. One of the important problems in

[1][Online]. Available: https://whisper2speech.github.io

developing whisper-to-speech VC systems is the lack of appropriate data for training. Usually, parallel whisper-voiced dataset or at least labeled whisper data is needed to train the whole system. While there are tools available for producing voice to whisper conversion, such pseudo-whispered data sounds unnatural.

A. WHISPER-TO-SPEECH VOICE CONVERSION

To authors knowledge, there is a lack of literature addressing to the whispered speech conversion problem, particularly in the context of streaming whisper-to-speech VC system. However, a variety of architectures was used for whisper-to-speech VC task earlier: Gaussian Mixture Models (GMM) [8], Sequence-to-Sequence models [9], Recurrent Neural Networks (RNN) [10] and different generative architectures. For instance, in [11] authors propose MelGAN [12] and VQ-VAE [13] based approaches adapted to whispered speech conversion. Parallel dataset of whispered and regular speech was used for the adaptation. Also the models objective functions were modified to fit the target conversion task. According to [11], the MelGAN based model outperforms VQ-VAE and DiscoGAN [14] based approaches.

The DiscoGAN model is also implemented in [15]. In this research the authors focus on improving F0 reconstruction using separate models for Mel-frequency Cepstral Coefficients (MFCC) and F0 prediction. However, the training scheme is organized in a 2-step way and the model for F0 prediction relies on the output from CycleGAN trained to reconstruct MFCC. This leads to noise in prediction and lack of speech naturalness in the final conversion. In order to solve this problem, in [16] propose to use CinC-Gan for more effective F0 prediction. The authors rely on simultaneously predicting F0 from the MFCC features of converted speech via joint training. According to them, such a scheme provides better quality both in objective and subjective terms relatively to the described CycleGAN based approach.

Alternatively, in [17] the authors rely on the power of SSL models and propose an encoder-decoder approach, where the encoder is a fine-tuned HuBERT model [7] and the decoder is a modified FastSpeech2 system [18]. Then a HI-FI GAN vocoder [19] is employed to convert the mel-spectograms to speech waveform. As it was demonstrated by the authors, the converted waveforms reveal a lower Word Error Rate (WER) compared to the original whisper waveforms. However, all tests were performed only on part of the WTimit dataset,[2] while another part was reserved for the training procedure. While the model presented in the paper intended to operate in real-time scenario, there is no information provided about its streaming capabilities.

B. SSL MODELS

Self-supervised learning models have gained significant attention in the recent research of acoustic representations for various tasks. One key advantage of these models is their ability to achieve excellent result in various tasks even with limited amounts of data through the fine-tuning procedure. Initially, SSL models were primarily used in speech recognition [6], [7]. Subsequently, similar approach was successfully extended to another speech processing tasks such as language, emotion, speaker recognition [20], [21] and VC [22]. While the real-time scenarios are highly important use cases for ASR models, the streaming scenario is often challenging for such models since SSL pre-training procedure is performed on full-length files without streaming mode adaptation.

C. STREAMING ASR MODEL

Usually, RNN or Convolution Neural Network (CNN) based models [23] are successfully used for real-time processing, but have limited speech recognition capabilities. In contrast, transformer based models show high quality speech recognition, but can degrade significantly in streaming conditions due to limited available context. Several methods were proposed to adapt transformer models to streaming conditions. In [24] Transformed Transducer models are suggested. The idea is to replace RNN-based encoder in the RNN-T architecture with transformer blocks and use context masking for transformer layers to simulate limited context in training mode. But this method still has significant restrictions, since the look-ahead context grows with the number of stacked transformer layers.

Another example of context masking proposed in the Conformer U2++ model [25] suggests a unified model for streaming and non-streaming conditions. The dynamic chunk training strategy used to achieve sufficient results both in streaming and non-streaming mode. The Augmented Memory (AM-TRF) approach proposed in [26] is built to reduce computation in processing of the long-range left context. However, proposed scheme is difficult to parallelize and has duplicate computations. The Emformer approach [27] is created to overcome these limitations. One of the key improvements is the caching mechanism to prevent extra computations for the left context block.

III. DATASETS

This section describes the datasets used to train various models in the whisper-to-speech VC pipeline, as well as the test datasets used to assess the quality of the resulting systems.

A. ENCODER TRAINING

All datasets used for the encoder model training and their description listed in Table 1. Whisper-DS is a dataset containing whispered speech collected from different sources. We increase the sampling probability for whispered speech data to 0.2 for each utterance in a batch during the training procedure. All audio transcriptions are preprocessed with text normalization tools and converted into phoneme transcription

[2][Online]. Available: http://www.isle.illinois.edu/sst/data/wTIMIT/

TABLE 1. Encoder Trainig Datasets

Dataset	Dur. (hours)	Condition	Domain
LibriSpeech [28]	928.5	regular voice	reading
MCV EN [29]	1481.7		
NISP [30]	31.4		
SLR70, SLR83 [3]	36.3		
People Speech [31]	2337.4		reading/ spontan.
Whisper-DS	92.0	whisper	reading/ spontan.

with the espeak backend.[4] Thus, the resulting vocabulary consists of 59 phonemes. We have found that using a phoneme vocabulary instead of a grapheme vocabulary for the Text-to-Speech system [32] improves the quality of whisper-to-voice conversion and apply this approach for training ASR systems in our subsequent experiments.

B. DECODER TRAINING

For the decoder training LibriTTS [33] and VCTK [34] corpora are used. The VCTK corpus includes data from 110 speakers with a total duration of 44 hours sampled at 48 kHz. The LibriTTS corpus consists of 585 hours of speech data from 2456 speakers sampled at 24 kHz. LibriTTS is a subset derived from the LibriSpeech dataset. We randomly choose 2% of all utterances for our validation subset and use the rest of the data for training.

We resample data to 22050 Hz and extract 80-dimensional log mel-spectrograms with a window size equal to 1024 samples and a hop size equal to 256. As these datasets do not contain any whispered speech samples, all data is converted to whisper with the Praat Toolkit [5] before encoder feature extraction. As it was noted in introduction, such toolkit is incapable of generating high-quality whisper audio. However, in the absence of a publicly available whisper-speech parallel dataset, we find such whisper data useful for our experiments.

C. VOCODER ADAPTATION

To adapt the system to single speakers we use one speaker from HI-FI TTS [35] for male and LJ Speech [36] for female voices, respectively. From HI-FI TTS we choose a speaker with the largest total duration of recorded speech (in clean environment). Finally, we use approximately 24 hours of data for each male and female voice, correspondingly. Here the train datasets are also converted to whisper with the Praat toolkit for feature extraction.

[3][Online]. Available: https://www.openslr.org/resources.php
[4][Online]. Available: https://github.com/bootphon/phonemizer
[5][Online]. Available: https://www.praatvocaltoolkit.com/whisper.html

D. TEST DATASETS

In this section we describe the datasets used both for ASR and VC systems evaluation. Various whispered and voiced datasets are used for testing purposes.

LibriSpeech test-other: This dataset contains recordings from 33 speakers with a total duration of 5.1 hours, collected from the LibriVox corpus. Although this dataset does not contain any whispered audio samples, we use it for overall validation of the results of the encoder models in different cases.

Chains dataset: This dataset contains recordings from 36 speakers collected in different conditions. In our experiments we use regular speech (Chains-S) and whispered speech (Chains-W) conditions. Each speaker reads a set of sentences in a regular speech and then reads the same sentences in a whispered speech. The total duration of used data is approximately 2.5 hours. We use all of this dataset to test the quality of our VC systems in various ways.

Spontan dataset: As there are no publicly available datasets which include spontaneous whispered speech, we have recorded and compiled the Spontan dataset. Our aim was to gather a test dataset that accurately reflects real-world scenarios. The resulting Spontan dataset contains 240 files with a total duration of approximately 1.1 hours. During the recording process, 6 speakers were asked 20 questions and their whisper responses were simultaneously recorded on microphones of different types. To increase the diversity of recorded data, we utilized 4 different types of microphones during recording — 2 microphones per speaker. We use this dataset for testing the quality of our VC systems.

IV. SYSTEM DESCRIPTION

Our system is based on the PPG-VC [37] approach with significant modifications of the encoder and decoder parts and adaptation of these parts to the streaming mode. In this approach, the data is processed sequentially. The source whispered speech is processed by encoder model, the received PPGs and speaker identity vector are input to the decoder model. Decoder converts the input features into log-mel spectrograms, which are further processed into regular speech using vocoder model.

A. STREAMING VOICE CONVERSION INFERENCE

To support low-latency whisper-to-speech VC, we implement chunk-wise inference for each model in the overall pipeline. Data is processed in chunks, which consist of the left context part, the center context part, and the look-ahead part. The left context contains cached past data and is useful for adding previous information to the model. Center context is a newly arrived part of data that will be converted into speech. The look-ahead is a small part of the latest arrived data, which can be discarded from post-processing, but is important for accuracy of center context processing. As models iteratively use the processed result of the previous pipeline model, the total

overall pipeline delay is a sum of the largest used center context part and all look-ahead parts. To reduce the overall time delay, we use the same chunk size for the encoder, decoder and vocoder pipeline models and close to zero look-ahead chunk part for the decoder and vocoder. The left context size, central context size, and look-ahead size used in this paper were chosen to maximize VC audio quality while maintaining low streaming delay and small redundant computations.

B. ENCODER

In our experiments two different checkpoints of the HuBERT Large model [7], which often used in VC tasks [17], [22], [38] are used. In most experiments, SSL pre-trained on the Libri-Light dataset and fine-tuned on LibriSpeech dataset model (HuBERT-LL-LS) are used, but we also experiment with only an SSL pre-trained model (HuBERT-LL).

1) STREAMING MODE ADAPTATION

At first, we experiment with the adaptation of the HuBERT model to streaming mode. For this, we propose two different methods. The first is based on the commonly used principle of masking attention context in the transformer layers of the acoustic model, where a combination of upper and lower triangular matrices is used. As a result of our research, we set the number of non-zero right diagonals to 40 tokens or 0.8 seconds and the number of non-zero left diagonals to 120 tokens or 2.4 seconds, which correspond to high full-length transcription quality and also decrease WER in streaming processing. Note, that the use of a triangular attention mask limits the receptive field of the first transformer blocks, however, this effect decreases with each subsequent transformer layer. We found this method not useful for fine-tuning of the HuBERT-LL-LS model directly, but it can be applied for the adaptation of HuBERT-LL model.

The second method, which can be named chunk-wise fine-tuning can be used for the adaptation of both HuBERT-LL and HuBERT-LL-LS models for the streaming mode. The main idea of this approach is to make inference and training passes more similar. To implement such a scheme we split the full waveform to small chunks, process each chunk separately and then concatenate computed logits to perform one backward pass. This decreases the receptive field for the convolution and transformer layers, allowing us to train the model on short chunks without speech recognition quality degradation and the necessity to build an alignment between transcription and speech. Schematically our approach is presented in Fig. 1.

Parameters of the experiments with the HuBERT-LL and HuBERT-LL-LS models streaming mode adaptation are described in the Table 3. During warp-up iteration are trained only on the new softmax matrix. After completing the full training process we additionally fine-tune the models for 15 K iterations using the chunk-wise approach with the chunk parameters chosen randomly for each iteration in specificated bounds.

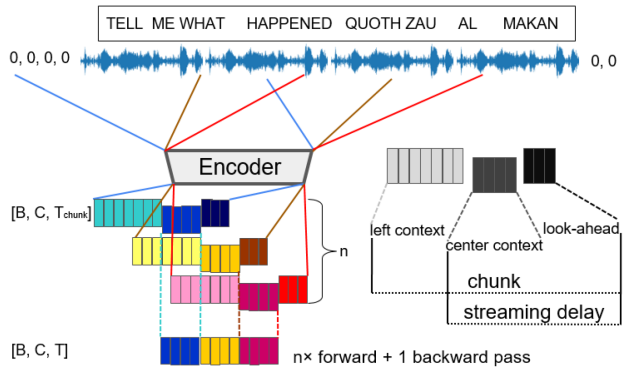

FIGURE 1. Chunk-wise adaptation of the ASR acoustic model for streaming mode. B,C and T are batch, channel and time feature dimension.

TABLE 2. WER Result on LibriSpeech Test-Other Dataset for Different Streaming Configuration,%

model	full file	2,0.4,0.4 s	1,0.3,0.2 s
HuBERT LL-LS	4.15	6.79	12.72
+ chunk-wise fine-tune	4.38	6.13	11.24
HuBERT LL + mask	4.95	6.69	11.71
+ chunk-wise fine-tune	4.97	6.34	9.30
Conformer U2++	8.43	11.04	13.03
EM-36L[27]	—	5.97	
Zipformer-T	6.36	7.42	7.88

The results of our experiment and their comparison with the EM-36L [27], Streaming Zipformer-T [6] and Conformer U2++ [7] streaming ASR models are described in Table 2. For the streaming case, the left context size, center context size, and look-ahead are equal to 2 s, 0.4 s, 0.4 s and 1 s, 0.3 s, 0.2 s, respectively. For EM-36 L we used one configuration where left context size, center context size, and look-ahead are 1.28 s, 0.64 s, 0.32 s. For Conformer U2++, these are 2.52 s, 0.84 s, 0 s and 1.08 s, 0.54 s, 0 s. For Streaming Zipformer-T, these are 3.2 s, 0.8 s, 0 s and 0.96 s, 0.48 s, 0 s. **Note, since the models architecture are different or publicly unavailable, nearest HuBERT streaming evaluation setting configuration in terms overall streaming delay and left context size is chosen for each model.**

Based on the presented results, it can be concluded that chunk-wise fine-tuning improves the quality of the ASR model in terms of WER on different streaming configurations with a slight decrease in WER on full-length files. Attention masking is useful too, but it decreases the quality of full-length speech recognition. Also, for the latency-tolerance streaming configuration, HuBERT might be a better choice. For further experiments we choose HuBERT LL-LS and streaming Zipformer-T models, and did not select the Emformer model due to lack the of an official checkpoint and difficulties in training the model.

[6][Online]. Available: https://huggingface.co/Zengwei/icefall-asr-librispeech-pruned- transducer-stateless7-streaming-2022-12-29

[7][Online]. Available: https://github.com/wenet-e2e/wenet/tree/main/examples/gigaspeech/s0

TABLE 3. Hyperparameters of Streaming Mode Adaptation and Whispered Domain Adaptation Experiments

Parameters	HuBERT-LL / HuBERT-LL-LS	HuBERT-LL-LS whisper adapt.
training data	LibriSpeech train	all train data
loss function	CTC loss	
warm-up iteration	2 k / —	2 k
training iteration	80 k	120 k
fine-tune iteration	15 k	15 k
left context size	0.8-2.0 s	0.8-2.0 s
center context size	0.3-0.4 s	0.4 s
look-ahead size	0.2-0.3 s	0.2-0.4 s
attention context	0.8 and 2.4 s / —	—
scheduler	triangle, max lr 4e-5	
optimizer	AdamW, betas 0.9,0.98	
batch size	220 s	440 s

2) WHISPERED DOMAIN ADAPTATION

In this part we present experiments with the adaptation of the model to the whispered domain. The parameters of the experiment are described in the Table 3. The HuBERT-LL-LS checkpoint with discarded linear head layer is used; for the warm-up iterations only the new softmax matrix is trained and then all layers except for the feature extractor are unfreezed for all the remaining iterations. We choose the best checkpoints in terms of Phoneme Error Rate (PER) on the Spontan and LibriSpeech test-other datasets for subsequent voice conversion experiments, discussed in Section V.

C. DECODER

We use the Tacotron 2 [39] model as a decoder with some changes from the baseline caused by differences in input features and the online attention mechanism. The ECAPA-TDNN [40] model is used for speaker identity extraction instead of the d-vector model because it provides better speaker identity quality compared to the d-vector model. The pitch processing part is discarded from the model due to a lack of pitch information in a whispered audio. Nearest interpolation is used to synchronize the time resolution between features at the decoder input and encoder output.

We also experiment with chunk-wise data preparation and extract features with the parameters for left context size, center context size, and look-ahead: 4 s, 0.4 s, and 0.4 s, instead of full-file feature extraction implemented in the original approach. The results of our experiments are described in Section V.

D. VOCODER

As a vocoder we use a pretrained HI-FI GAN model [19]. We fine-tune the base model using log-mel spectrograms predicted by a trained decoder to adapt the vocoder to such features. Only single-speaker datasets are used during fine-tuning, thus our final model introduces any-to-one VC.

TABLE 4. Comparison of Speech Recognition Quality by WER Between Encoders, %

model	Spontan	Chains-W	Chains-S
HuBERT-LL-LS	27.65	11.98	6.92
Whisper-L[41]	10.43	4.21	2.21
Conformer-T-L[42]	15.94	8.22	5.16
Zipformer-T	39.73	19.48	10.79

E. EVALUATION METRICS

For ASR encoders evaluation experiments we use WER and PER metrics. To compare audio outputs from voice conversion systems we use both objective and subjective metrics. As a subjective, a widespread Mean Opinion Score (MOS) measure is used. The process of MOS [12], [17] evaluation is designed as follows. We randomly choose 10 files from the Spontan and Chains-W datasets and prepare outputs from several conversion systems. Each utterance was evaluated by 20 volunteers on a 1–5 scale, where 1 is the worst quality of audio and 5 is the best. Participants were asked to measure naturalness and clarity of given recordings. As an objective metric for the conversion system, WER computed with the Conformer-Transducer-Large model for English ASR (Conformer-T-L) [8] is used. Note, that before WER calculation for whisper-to-speech VC on Spontan and Chains-W datasets, we first convert the original whispered speech of different speakers into the regular speech of one speaker.

V. EXPERIMENTS

In this section we describe our main results. Firstly, we compare the quality of different ASR models on whispered and regular speech in terms of WER. For this evaluation the official pretrained checkpoints are used. The results are presented in Table 4. Significant quality degradation is seen on whispered speech data in contrast to regular speech when comparing the WER metric on the speech part of the Chains dataset (Chains-S) and whispered one (Chains-W). Also, remarkable differences in the quality of described encoders in general are observed. For instance, the Whisper-L model outperforms HuBERT-LL-LS model more than twice. However, using the best ASR encoder in terms of WER on the speech recognition task does not necessarily lead to getting the best whisper-to-speech VC system.

Next, we compare the quality of the full whisper-to-speech VC system in terms of WER. The comparison of systems based on different encoders is presented in Table 5 for the Spontan and Chains-W datasets. In these experiments PPGs (for the HuBERT-LL-LS encoder) and embeddings from the last encoder layer (for Whisper-L, Zipformer-T, Conformer-T-L encoders) are used for training decoder model. All other hyper-parameters were similar at time of decoder training. The system based on the HuBERT-LL-LS encoder performs better on the Chains-W dataset and has similar results with

[8][Online]. Available: https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models /stt_en_conformer_transducer_large

TABLE 5. WER After Whisper-to-Speech VC on the Spontan and Chains-W Test Sets Computed With Conformer-T-L,%. For the Case of Streaming Left Context Size, Center Context Size, and Look-Ahead are 2 s, 0.4 s, 0.4 s and 1 s, 0.4 s, 0.3 s, Respectively

model	full file	2 s, 0.4 s	1 s, 0.3 s
Spontan			
Whisper-L + PPG-VC	24.18	—	—
Conformer-T-L + PPG-VC	36.24	—	—
Zipformer-T + PPG-VC	44.43	48.52	49.92
HuBERT-LL-LS + PPG-VC	25.39	40.42	43.79
+ chunk-wise decoder train.	24.61	34.06	39.49
+ whisper adaptation	14.43	21.39	27.24
+ chunk-wise encoder train.	**14.35**	**19.18**	**21.28**
orignal PPG [37]	58.48	—	—
MelGAN [11]	89.21	—	—
Chains-W			
Whisper-L + PPG-VC	14.87	—	—
Conformer-T-L + PPG-VC	17.97	—	—
Zipformer-T + PPG-VC	21.62	23.74	23.08
HuBERT-LL-LS + PPG-VC	9.68	12.83	15.21
+ chunk-wise decoder train.	9.53	12.48	14.42
+ whisper adaptation	8.53	10.48	12.02
+ chunk-wise encoder train.	**8.46**	**9.40**	**10.61**
orignal PPG-VC [37]	32.06	—	—
MelGAN [11]	56.80	—	—

The better result is marked in bold.

Whisper-L on the Spontan dataset. We choice HuBERT-LL-LS encoder as our baseline system for other experiment. We did not select the Whisper-L model because the base architecture cannot be used for stream processing without excessive computational cost, and the commonly used speedup techniques (VAD, batching) [43] are not applicable to the streaming VC task. Although the Zipformer-T based system has good streaming performance, we did not select this due to significantly high WER values in whisper-to-speech VC.

We propose several improvements to the training scheme of the baseline system. Firstly, we prepare encoder features for the decoder training in chunk mode described in Section IV-C and get an improvement in terms of WER for streaming whisper-to-speech VC. Secondly, adaptation to the whispered speech domain for encoder described in Section IV-B2 is used which improves the quality of whisper-to-speech VC for all test cases. We also observe more significant improvements for the Spontan dataset which are due to spontaneous whispered data additionally used during training. Thirdly, we apply a chunk-wise fine-tuning approach, described in Section IV-B1 for encoder training and get improvement in terms of WER in various streaming configurations. And finally, we compare our models with the default PPG-VC model and MelGAN approach described in Section II-A. According to the presented results our best system significantly outperforms the original PPG-VC and MelGAN approaches. We present all results of the described experiments in Table 5.

TABLE 6. MOS Evaluation Results With 95% Confidence Interval for the Chains-W and Spontan Datasets

model	Spontan	Chains-W
Ground true whisper	3.31 ± 0.18	3.47 ± 0.19
original PPG-VC	1.90 ± 0.16	2.90 ± 0.15
HuBERT-LL-LS + PPG-VC	3.60 ± 0.16	4.04 ± 0.14
+ streaming inference	3.11 ± 0.15	$\mathbf{4.11} \pm 0.14$
+ whsp adapt., chunk-wise train.	$\mathbf{4.25} \pm 0.11$	3.96 ± 0.13
+ streaming inference	4.07 ± 0.13	4.07 ± 0.15

The better result is marked in bold.

In Table 6 we present the outcomes of MOS evaluation. As it is found the results of MOS assessment mostly correlate with objective evaluation. The original PPG-VC model, which performs poorly in terms of WER metric, also has a lower MOS value. It can be seen that the system based on a chunk-wise fine-tuned encoder adapted to whispered speech shows a higher MOS value than the system based on the original HuBERT-LL-LS on the complicated Spontan dataset.

VI. DISCUSSION

Despite a significant increase in audio quality of the whisper-to-speech VC using the chunk-wise training approach for HuBERT-LL-LS model, some issues remain unresolved. Firstly, the values of WER on the initial whisper data and after conversion on Chains-W are comparable, but inferior to the quality of the original speech recognition on Chains-S presented in Tables 4, 5. This indicates that the conversion result is still worse than speech. Secondly, we are currently unable to take advantage of joint encoder-decoder ASR models such as transducers, seq2seq approaches, etc. These models typically outperform the single encoder model in terms of WER, but transferring this improvement to conversion may be challenging due to the time resolution mismatch between original audio and model features. To further improve the audio quality of low-delayed conversion, experiments with the adaptation ASR model to streaming settings seem important. Also, the problem of any2any whisper-to-speech VC is unresolved in our paper, despite the importance of maintaining the speaker identity for comfortable communication between people.

VII. CONCLUSION

In this paper, a method of low latency whisper-to-speech voice conversion was proposed. Several approaches to adapt HuBERT model to streaming settings were considered, and their influence on the quality of low-delayed speech recognition and voice conversion was shown. As a result, the audio quality of the whisper-to-speech VC was significantly improved. This improvement was obtained in processing of full files as well as in streaming mode scenario and compared with the available baseline systems on whispered speech datasets.

TABLE 7. Comparison of ASR Quality With Google Cloud STT After Whisper-to-Speech Voice Conversion (W2S VC) With Our Model and WESPER System on the Wtimit Test Set

Google Cloud STT	speech type	WER	CER
[17]	normal	11.55	4.66
	whisper	44.70	28.38
	WESPER W2S VC	26.68	12.70
default	normal	16.81	7.45
	whisper	52.83	34.98
	our W2S VC	20.00	9.17
latest short	normal	10.57	4.25
	whisper	28.01	15.49
	our W2S VC	19.21	8.81

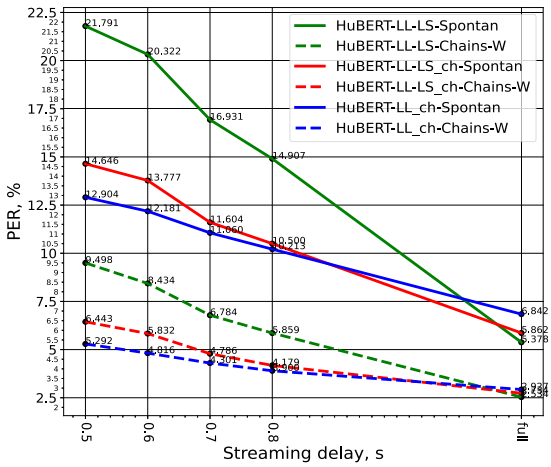

FIGURE 2. PER on Spontan and Chains-W test sets for proposed streaming and whisper mode adapted ASR model. Streaming delay is sum of center context and look-ahead; left context set to 2 s, "full" corresponds to the processing of files entire.

APPENDIX A

A1. STREAMING ASR MODELS FOR CONVERSION

In this additional study, we explore in more detail the impact of streaming encoder speech recognition quality on real-time whisper-to-speech VC audio quality. The whisper-to-speech VC system fine-tuned into whisper domain models are used. We compare the encoder models trained for full file processing (HuBERT-LL-LS) with the ones adapted for streaming mode with chunk-wise (HuBERT-LL-LS-ch) and attention context masking with chunk-wise (HuBERT-LL-ch) methods. First, we compare only encoder models in terms of PER on Chains-W and Spontan datasets. Results are shown in Fig. 2. We can see that the degradation in PER for more complex streaming cases is reduced when using the proposed streaming methods, especially attention context masking with chunk-wise approach. However, in Table 8 the best conversion quality in terms of WER is achieved when using into VC pipeline an encoder that is not the best for streaming. Based on the obtained results, it can be assumed that a good quality of speech recognition on full files and least degradation in streaming mode both are necessary for the best audio quality of VC. However, this result requires further research of various combinations of encoders and decoders in one conversion pipeline for different training approaches.

TABLE 8. WER After Whisper-to-Speech VC on Spontan and Chains-W Test Sets Computed With Conformer-T-L,%. For the Case of Streaming Left Context Size, Center Context Size, and Look-Ahead are 2 s, 0.4 s, 0.4 s and 1 s, 0.4 s, 0.3 s

model	full file	2 s, 0.4 s	1 s, 0.3 s
Spontan			
HuBERT-LL-LS	14.43	21.39	27.24
HuBERT-LL-ch	17.13	19.35	24.02
HuBERT-LL-LS-ch	14.35	19.18	21.28
Chains-W			
HuBERT-LL-LS	8.53	10.48	12.02
HuBERT-LL-ch	8.76	9.73	12.13
HuBERT-LL-LS-ch	8.46	9.40	10.61

A2. COMPARISON WITH THE WESPER SYSTEM

In this additional study, we compare our results with results of the WESPER model proposed in [17]. Evaluation is done on the wTimit dataset. In this dataset 48 speakers were asked to pronounce prompts from the TIMIT dataset in normal and whispered speech. Due to the lack of detailed information about the test settings in [17] the WESPER results are not reproducible. Thus, we evaluate our system on the wTimit-test dataset with "default" and "latest short" Google Cloud STT models.[9] A comparison of this evaluation with results presented in [17] is given in Table 7. Although we cannot directly compare the systems, according to the tests, our system performs better even though the WESPER model was pretrained on the training part of wTimit dataset.

REFERENCES

[1] S. A Sheikh, Md Sahidullah, F. Hirsch, and S. Ouni, "StutterNet: Stuttering detection using time delay neural network," in *Proc. 29th Eur. Signal Process. Conf.*, 2021, pp. 426–430.
[2] X. Zhang et al., "Stutter-TTS: Controlled synthesis and improved recognition of stuttered speech," 2022, *arXiv:2211.09731v1*.
[3] R. J. Ingham, A. K. Bothe, E. Jang, L. Yates, J. Cotton, and I. Seybold, "Measurement of speech effort during fluency-inducing conditions in adults who do and do not stutter," *J. Speech, Lang., Hear. Res.*, vol. 52, no. 5, pp. 1286–301, 2009.
[4] D. T. Grozdic and S. T. Jovicic, "Whispered speech recognition using deep denoising autoencoder and inverse filtering," *IEEE/ACM Trans. Audio, Speech, Lang, Process.*, vol. 25, pp. 2313–2322, Dec. 2017.
[5] H.-J. Chang, A. H. Liu, H. y. Lee, and L.-S. Lee, "End-to-end whispered speech recognition with frequency-weighted approaches and pseudo whisper pre-training," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2020, pp. 186–193.
[6] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 12449–12460.
[7] 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, Lang, Process.*, vol. 29, pp. 3451–3460, 2021.
[8] T. Toda and K. Shikano, "NAM-to-Speech conversion with Gaussian mixture models," in *Pro. 9th Eur. Conf. Speech Commun. Technol.*, 2005, pp. 1957–1960.

[9][Online]. Available: https://cloud.google.com/speech-to-text, date of the application 28.08.23

[9] H. Lian, Y. Hu, W. Yu, J. Zhou, and W. Zheng, "Whisper to normal speech conversion using sequence-to-sequence mapping model with auditory attention," *IEEE Access*, vol. 7, pp. 130495–130504, 2019.

[10] N. Meenakshi and P. K. Ghosh, "Whispered speech to neutral speech conversion using bidirectional LSTMs," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 491–495.

[11] D. Wagner, S. P. Bayerl, A. HéctorC. Maruri, and T. Bocklet, "Generative models for improved naturalness, intelligibility, and voicing of whispered speech," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2022, pp. 943–948.

[12] K. Kumar et al., "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 14910–14921.

[13] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6309–6318.

[14] T. Kim, M. Cha, H. Kim, Jung Kwon Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1857–1865.

[15] M. Parmar, S. Doshi, Nirmesh J. Shah, M. Patel, and H. A. Patil, "Effectiveness of cross-domain architectures for whisper-to-normal speech conversion," in *Proc. 27th Eur. Signal Process. Conf.*, 2019, pp. 1–5.

[16] M. Patel, M. Purohit, J. Shah, and H. A. Patil, "CINC-GAN for effective F0 prediction for whisper-to-normal speech conversion," in *Proc. 28th Eur. Signal Process. Conf.*, 2020, pp. 411–415.

[17] J. Rekimoto, "WESPER: Zero-shot and realtime whisper to normal voice conversion for whisper-based speech interactions," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2023, pp. 1–12.

[18] Y. Ren et al., "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 8588–8592.

[19] J. Kong, J. Kim, and J. Bae, "HIFI-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 17022–17033.

[20] L. Pepino, P. Riera, and L. Ferrer, "Emotion recognition from speech using wav2vec 2.0 embeddings," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 3400–3404.

[21] N. Vaessen and D. A. van Leeuwen, "Fine-tuning wav2vec2 for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 7967–7971.

[22] B. van Niekerk, M.-A. J. Carbonneau, M. Zaïdi, H. Baas Seuté, and H. Kamper, "A comparison of discrete and soft speech units for improved voice conversion," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, 2022, pp. 6562–6566.

[23] T. N. Sainath et al., "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 6059–6063.

[24] Q. Zhang et al., "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7829–7833.

[25] B. Zhang et al., "Unified streaming and non-streaming two-pass end-to-end model for speech recognition," 2012, *arXiv:2012.05481v2*.

[26] C. Wu, Y. Wang, Y. Shi, C.-F. Yeh, and F. Zhang, "Streaming transformer-based acoustic models using self-attention with augmented memory," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 2132–2136.

[27] S. Yangyang et al., "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 6783–6787.

[28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 5206–5210.

[29] R. Ardila et al., "Common voice: A massively-multilingual speech corpus," in *Proc. 12th Conf. Lang. Resour. Eval.*, 2020, pp. 4211–4215.

[30] S. B. Kalluri, D. Vijayasenan, S. Ganapathy, R. R. M, and P. Krishnan, "NISP: A multi-lingual multi-accent dataset for speaker profiling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 6953–6957.

[31] D. Galvez et al., "The people's speech: A large-scale diverse english speech recognition dataset for commercial usage," in *Proc. 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track (Round 1)*, 2021.

[32] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 5530–5540.

[33] H. Zen et al., "LibriTTS: A. Corpus Derived from LibriSpeech for Text-to-Speech," in *Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 1526–1530.

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

[35] E. Bakhturina, V. Lavrukhin, B. Ginsburg, and Y. Zhang, "Hi-Fi multi-speaker english TTS dataset," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2776–2780.

[36] K. Ito and L. Johnson, "The LJ speech dataset," 2017. [Online]. Available: https://keithito.com/LJ-Speech-Dataset/

[37] S. Liu, Y. Cao, D. Wang, X. Wu, X. Liu, and Helen M. Meng, "Any-to-many voice conversion with location-relative sequence-to-sequence modeling," *IEEE/ACM Trans. Audio,Speech, Lang. Process.*, vol. 29, pp. 1717–1728, 2021.

[38] H. Guo, C. Liu, Carlos Toshinori Ishi, and H. Ishiguro, "QuickVC: Any-to-many voice conversion using inverse short-time fourier transform for faster conversion," 2023, *arXiv:2302.08296v4*.

[39] J. Shen et al., "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4779–4783.

[40] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 3830–3834.

[41] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. 40th Int. Conf. Mach. Learn.*, 2023, pp. 28492–28518.

[42] G. Anmol et al., "Conformer: Convolution-augmented transformer for speech recognition," 2020, *arXiv:2005.08100v1*.

[43] M. Bain, J. Huh, T. Han, and A. Zisserman, "WhisperX: Time-accurate speech transcription of long-form audio," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2023, pp. 4489–4493.