ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9746484

A COMPARISON OF DISCRETE AND SOFT SPEECH UNITS FOR IMPROVED VOICE CONVERSION

Benjamin van Niekerk[1,2], *Marc-André Carbonneau*[1], *Julian Zaïdi*[1], *Matthew Baas*[2], *Hugo Seuté*[1], *Herman Kamper*[2]

[1]Ubisoft La Forge, Montreal, Canada
[2]E&E Engineering, Stellenbosch University, South Africa

ABSTRACT

The goal of voice conversion is to transform source speech into a target voice, keeping the content unchanged. In this paper, we focus on self-supervised representation learning for voice conversion. Specifically, we compare discrete and soft speech units as input features. We find that discrete representations effectively remove speaker information but discard some linguistic content – leading to mispronunciations. As a solution, we propose soft speech units learned by predicting a distribution over the discrete units. By modeling uncertainty, soft units capture more content information, improving the intelligibility and naturalness of converted speech.[1][2]

***Index Terms*—** voice conversion, speech synthesis, self-supervised learning, acoustic unit discovery

1. INTRODUCTION

Voice conversion systems transform source speech into a target voice, keeping the content unchanged. From re-creating young Luke Skywalker in The Mandalorian [1], to restoring the voice of an Amyloidosis patient [2], voice conversion has applications across entertainment, education and healthcare.

In a typical voice conversion system, the goal is to learn features that capture linguistic content but discard speaker-specific details. We can then replace the speaker information to synthesize audio in a target voice. While systems trained on parallel data [3, 4] or text transcriptions [5, 6] produce convincing results, they require costly data collection and annotation efforts. Unsupervised voice conversion addresses this issue by learning without labels or parallel speech [7, 8]. However, there is still a gap in quality and intelligibility between unsupervised and supervised systems [9].

To bridge this gap, recent work investigates self-supervised representation learning for voice-conversion. Most of these studies focus on discrete speech units [10–12]. The idea is that discretization imposes an information bottleneck separating content from speaker details. While effective at removing speaker information, discretization also discards some linguistic content – increasing mispronunciations in the converted speech. Take the word *fin*, for example. Ambiguous frames in the fricative /f/ may be assigned to incorrect nearby units, resulting in the mispronunciation *thin*.

To tackle this problem, we propose soft speech units. Using a fine-tuning procedure similar to [13], we train a network to predict a distribution over discrete speech units. By modeling uncertainty in discrete-unit assignments, we aim to retain more content information and, as a result, correct mispronunciations like *fin-thin*. This idea is inspired by soft-assignment in computer vision, which has been shown to improve performance on classification tasks [14].

Focusing on any-to-one voice conversion (i.e., any source speaker to a single target speaker), we compare discrete and soft speech units across two self-supervised methods: contrastive predictive coding (CPC) [15] and hidden-unit BERT (HuBERT) [13]. Finally, we evaluate discrete and soft units on a cross-lingual voice conversion task.

Our main contributions are as follows:

- We propose soft speech units for voice conversion and describe a method to learn them from discrete units.
- We find that soft units improve intelligibility and naturalness compared to discrete speech units.
- We show that soft units transfer better to unseen languages in cross-lingual voice conversion.

2. VOICE CONVERSION SYSTEMS

In this section, we describe the voice conversion system we use to compare discrete and soft speech units. Figure 1 shows an overview of the architecture. The system consists of three components: a content encoder, an acoustic model, and a vocoder. The *content encoder* extracts discrete or soft speech units from input audio (illustrated in Figure 1a and 1b, respectively). Next, the *acoustic model* translates the speech units into a target spectrogram. Finally, the spectrogram is converted into an audio waveform by the *vocoder*.

[1]Audio samples available at `https://ubisoft-laforge.github.io/speech/soft-vc/`

[2]Code available at `https://github.com/bshall/soft-vc`

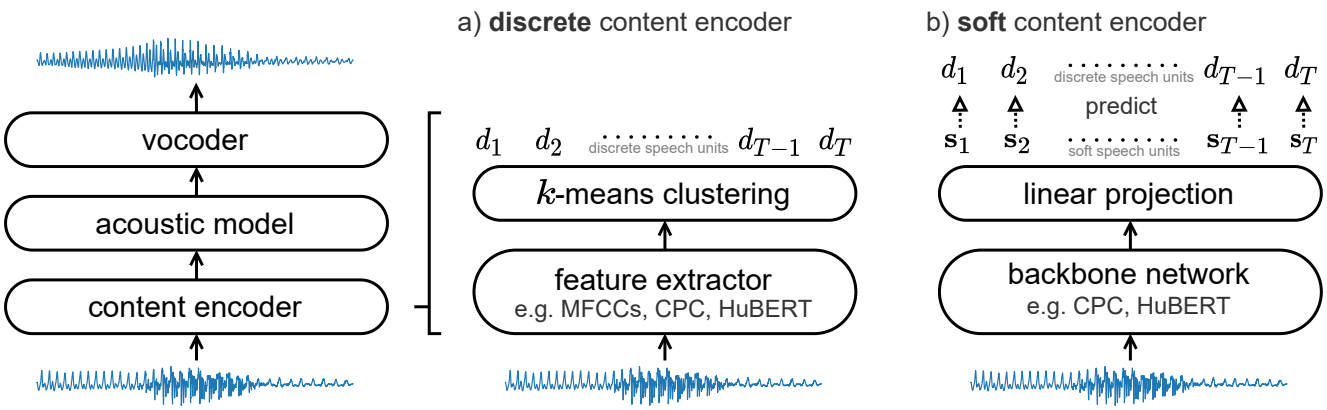

Fig. 1. Architecture of the voice conversion system. a) The **discrete** *content encoder* clusters audio features to produce a sequence of discrete speech units. b) The **soft** *content encoder* is trained to predict the discrete units. The *acoustic model* transforms the discrete/soft speech units into a target spectrogram. The *vocoder* converts the spectrogram into an audio waveform.

2.1. Content Encoders

Discrete Content Encoder: The discrete content encoder consists of feature extraction followed by k-means clustering (see Figure 1a). Different feature extractors can be used in the first step – from low level descriptors such as MFCCs to self-supervised models like CPC or HuBERT. In the second step, we cluster the features to construct a dictionary of discrete speech units. Previous work shows that clustering features from large self-supervised models improves unit quality [13, 16, 17] and voice conversion [10]. Altogether, the discrete content encoder maps an input utterance to a sequence of discrete speech units $\langle d_1, \ldots, d_T \rangle$.

Soft Content Encoder: For soft speech units, it is tempting to directly use the output of the feature extractor without clustering. However, previous work [17, 18] shows that these representations contain large amounts of speaker information, rendering them unsuitable for voice conversion (we confirm this in our experiments later). Instead, we train the soft content encoder to predict a distribution over discrete units.

The idea is that soft speech units provide a middle-ground between raw continuous features and discrete units. On the one hand, discrete units create an information bottleneck that forces out speaker information. So to accurately predict the discrete units, the soft content encoder needs to learn a speaker independent representation. On the other hand, the space of speech sounds is not discrete. As a result, discretization causes some loss of content information. By modeling a distribution over discrete units, we aim to keep more of the content information and increase intelligibility.

Figure 1b outlines the training procedure for the soft content encoder. Given an input utterance, we first extract a sequence of discrete speech units $\langle d_1, \ldots, d_T \rangle$ as labels. Next, a backbone network (e.g., CPC or HuBERT) processes the utterance. Then, a linear layer projects the outputs to produce a sequence of soft speech units $\langle \mathbf{s}_1, \ldots, \mathbf{s}_T \rangle$. Each soft unit parameterizes a distribution over the dictionary of discrete units:

$$p(d_t = i \mid \mathbf{s}_t) = \frac{\exp(\text{sim}(\mathbf{s}_t, \mathbf{e}_i)/\tau)}{\sum_{k=1}^{K} \exp(\text{sim}(\mathbf{s}_t, \mathbf{e}_k)/\tau)},$$

where i is the cluster index of the i^{th} discrete unit, $\mathbf{e}_i$ is a corresponding trainable embedding vector, $\text{sim}(\cdot, \cdot)$ computes the cosine similarity between the soft and discrete units, and τ is a temperature parameter. Finally, we minimize the average cross-entropy between the distributions and discrete targets $\langle d_1, \ldots, d_T \rangle$ to update the encoder (including the backbone). At test time, the soft content encoder maps input audio to a sequence of soft speech units $\langle \mathbf{s}_1, \ldots, \mathbf{s}_T \rangle$, which is then passed on to the acoustic model.

2.2. Acoustic Model and Vocoder

The acoustic model and vocoder are typical components in a text-to-speech (TTS) system, e.g., [19, 20]. For voice conversion, the inputs to the acoustic model are speech units rather than graphemes or phonemes. The acoustic model translates the speech units (either discrete or soft) into a spectrogram for the target speaker. Then, the vocoder converts the predicted spectrogram into audio. There are a range of options for high-fidelity vocoders, including WaveNet [21] and HiFi-GAN [22].

3. EXPERIMENTAL SETUP

We focus on any-to-one conversion using LJSpeech [23] for the target speaker. We compare discrete and soft speech units across two tasks: intra- and cross-lingual voice conversion. For intra-lingual conversion, we use the LibriSpeech [24] dev-clean set as source speech. In the cross-lingual task, we apply systems trained on English to French and Afrikaans data. For French, we use the CSS10 dataset [25]. For Afrikaans, we use data from the South African languages corpus [26].

3.1. Model Implementation

To compare discrete and soft speech units, we implement different versions of the voice conversion system described in Section 2. For the discrete content encoder, we test CPC and HuBERT as feature extractors. Similarly, in the soft content encoder, we evaluate CPC and HuBERT as backbones.

CPC learns linguistic representations by predicting future audio segments from a given context. Based on a contrastive loss, the goal is to distinguish correct future frames from negative examples drawn from other audio files. We use CPC-big[3] [16] pretrained on the LibriLight unlab-6k set [27].

HuBERT consists of two steps: acoustic unit discovery followed by masked prediction. The first step is to construct labels for the prediction task by clustering either low-level speech features or learned representations from previous training iterations. The second step is to predict these labels for masked spans of input audio. We use HuBERT-Base[4] [13] pretrained on LibriSpeech-960 [24].

Discrete Content Encoder: To learn discrete speech units, we apply k-means clustering to intermediate representations from CPC-big or HuBERT-base. We use 100 clusters and estimate their means on a subset of 100 speakers from the LibriSpeech train-clean-100 split. For CPC-big we cluster the outputs of the second LSTM layer in the context network. Additionally, we apply the speaker normalization step proposed in [17]. For HuBERT-Base, we use the the seventh transformer layer. We choose these layers because the resulting acoustic units perform well on phone discrimination tests [13, 16, 17].

Soft Content Encoder: For the soft content encoder, we use CPC-big and HuBERT-Base as backbones. We fine-tune each model (including the backbone) on LibriSpeech-960 to predict the corresponding discrete speech units. We train for 25k steps using a learning rate of 2×10^{-5}.

Acoustic Model and Vocoder: The acoustic model maps speech units (discrete or soft) to a target spectrogram. The structure of the model is based on Tacotron 2 [19] and consists of an encoder and autoregressive decoder. The encoder is built from a pre-net (two linear layers with dropout), followed by a stack of three 1D-convolutional layers with instance normalization. For discrete units, we use an initial embedding table to map cluster indexes to vectors. The decoder predicts each spectrogram frame from the outputs of the encoder and past frames. We apply a pre-net to the previously predicted frame and concatenate the result with the output of the encoder. Three LSTM layers with residual connections are then used to model long-term dependencies. Finally, a linear layer predicts the next spectrogram frame.

The vocoder turns the spectrogram frames into an audio waveform. We choose HiFi-GAN as the vocoder since several papers show that it produces high-quality speech [10, 20].

The acoustic model and vocoder are trained on LJSpeech. We downsample the dataset to 16 kHz and extract 128-band mel-spectrograms at a hop-length of 10 ms with a Hann window of 64 ms. We train the acoustic model for 50k steps and select the checkpoint with the lowest validation loss. For HiFi-GAN, we train using ground-truth spectrograms for 1M steps and then fine-tune on predicted spectrograms for 500k steps.

3.2. Baseline Models

We also compare our voice conversion systems against two common baselines: AutoVC[5] [8] and the Cascaded ASR-TTS[6] system [6] from the Voice Conversion Challenge 2020 [9]. AutoVC is an any-to-any voice conversion system based on an autoencoder with a down-sampling bottleneck. For the Cascaded ASR-TTS model, input speech is first transcribed using an automatic speech recognition (ASR) system. The transcripts are then piped to a text-to-speech (TTS) system which generates audio in the target voice. We fine-tune the released Cascade ASR-TTS checkpoint on LJSpeech.

3.3. Evaluation Metrics

To assess the *intelligibility* of the converted speech, we measure word error rate (WER) and phoneme error rate (PER) using an automatic speech recognition (ASR) system. We use the HuBERT-Large ASR[7] model [13] for orthographic transcriptions, and Allosaurus [28] for phonemic transcriptions. We convert the ground-truth orthographic transcriptions to phonemes using epitran [29]. Lower error rates correspond to more intelligible speech since it shows that the original words are still recognizable after conversion. We use Google Cloud Speech-to-Text for Afrikaans ASR and Wav2vec2 for French.

We measure *speaker-similarity* using a trained speaker-verification system. Given a query and enrollment utterance, the verification system assigns a similarity score indicating whether the speakers match or not. We use cosine similarity between x-vectors [30] as the score. For evaluation, each converted example is paired with 50 different enrollment utterances sampled from the target speaker. Then we add an equal number of authentic target speaker pairs. We report equal-error rate (EER), which approaches 50% when the verification system cannot distinguish between converted and genuine target-speaker utterances (indicating high speaker similarity).

For *naturalness*, we conduct a subjective evaluation based on mean opinion scores (MOS). We randomly select 20 source speakers from the LibriSpeech dev-clean set and convert 3 utterances per speaker, each between 4 and 15 seconds long. Evaluators rate the naturalness of the utterances on a five-point scale (from 1=bad to 5=excellent). Each sample is judged by at least 4 self-reported English speakers using the CAQE Toolkit [31]. We report MOS and 95% confidence intervals.

[3] https://github.com/facebookresearch/CPC_audio
[4] https://github.com/pytorch/fairseq
[5] https://github.com/auspicious3000/autovc
[6] https://github.com/espnet/espnet/tree/master/egs/vcc20
[7] https://github.com/huggingface/transformers

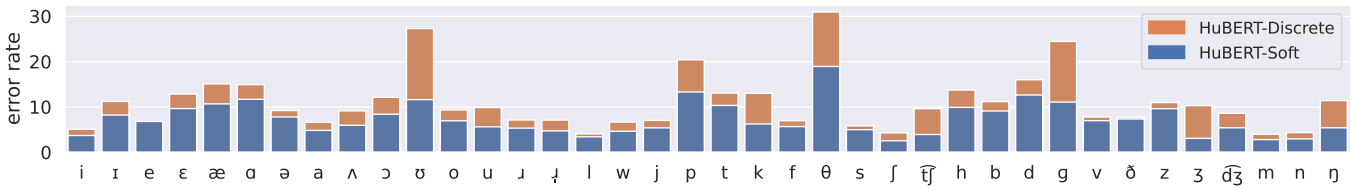

Fig. 2. Breakdown of PER (%) per phoneme for HuBERT-Discrete and HuBERT-Soft.

4. EXPERIMENTAL RESULTS

We report results for the English intra-lingual experiments followed by the French and Afrikaans cross-lingual results.

Intelligibility: Table 1 reports intelligibility in terms of PER and WER, with lower rates indicating more intelligible speech. Compared to discrete units, using soft speech units substantially improves WER (by around 50% relative) and PER (by over 20%). Both Hubert- and CPC-Soft outperform the cascaded ASR-TTS baseline, reaching error rates close to the ground-truth recordings.

Figure 2 compares the PER between soft and discrete units in more detail. We align phonemic transcriptions of converted speech with the ground-truth and compute an error rate for each phoneme. Apart from the vowel /ʊ/ (c**oo**k), improvements are primarily in the consonants. In particular, we see marked improvement in the affricative / ʧ/ (**ch**in), the fricative /ʒ/ (**j**oke), and the velar stops /k/ (**k**id) and /g/ (**g**o).

To summarize, the results show that soft assignments capture more linguistic content, improving intelligibility compared to discrete units.

Method	PER	WER	EER	MOS
HuBERT-Discrete	10.4	5.4	49.8	3.69 ± 0.13
HuBERT-Soft	**7.8**	**2.6**	45.6	$\mathbf{4.15 \pm 0.12}$
HuBERT-Raw-Features	11.3	2.8	27.8	-
CPC-Discrete	14.5	8.1	**50.0**	3.37 ± 0.13
CPC-Soft	11.4	3.7	41.3	3.91 ± 0.12
CPC-Raw-Features	14.6	3.6	5.2	-
AutoVC [8]	58.3	73.3	13.3	1.09 ± 0.04
Cascaded ASR-TTS [6]	8.4	7.4	46.8	3.15 ± 0.11
Ground Truth	7.9	2.0	-	4.57 ± 0.10

Table 1. Voice conversion results on English. PER (%) and WER (%) assess intelligibility. Speaker-similarity is reported as the EER (%) of a speaker-verification system. MOS from subjective tests indicate naturalness.

	French		Afrikaans	
Method	WER	EER	WER	EER
HuBERT-Discrete	64.6	49.6	24.7	37.6
HuBERT-Soft	28.2	33.9	12.9	28.2
Ground Truth	14.6	-	8.0	-

Table 2. Intelligibility (WER) and speaker similarity (EER) results (%) for the cross-lingual experiments.

Speaker Similarity: The third column of Table 2 shows speaker similarity results. An EER of 50% demonstrates that the speaker verification system cannot differentiate between genuine and converted utterances (i.e., high speaker similarity). Discrete units obtain near-perfect scores, verifying that they effectively discard source-speaker information. In comparison, soft units cause a small decrease in similarity. However, raw features are notably worse, confirming that soft units are a good middle-ground between discrete and continuous features.

Naturalness: The last column of Table 2 reports MOS for naturalness. Across both the CPC- and HuBERT-based models, soft units significantly improve over discrete speech units. Hubert-Soft performs best in the listening test, with naturalness scores approaching the ground-truth recordings. We speculate that better intelligibility explains part of the improvement. However, we also think that additional information encoded in the soft units results in more natural prosody. This could explain why the Cascaded ASR-TTS system scores lower on naturalness despite a better PER and WER than CPC-Discrete.

Cross-Lingual Voice Conversion: Table 2 reports intelligibility and speaker-similarity scores for the cross-lingual task. We take the best discrete and soft systems from the intra-lingual setting above, and apply them directly to French and Afrikaans test data. Comparing WER results, we see that soft speech units transfer better to unseen languages. However, in the cross-lingual setting, soft units lead to a larger drop in speaker similarity (lower EER). We suspect this is because soft units retain more accent information from the source speech.

5. CONCLUSION

We proposed soft speech units to improve unsupervised voice conversion. We showed that soft units are a good middle-ground between discrete and continuous features – they accurately represent linguistic content while still discarding speaker information. In both objective and subjective evaluations, we found that soft units improve intelligibility and naturalness. Future work will investigate soft speech units for any-to-any voice conversion.

6. REFERENCES

[1] "Making of season 2 finale," *Disney Gallery: The Mandalorian*, 2021.

[2] Greg Singer, "Respeecher gives voice to Michael York in healthcare initiative," `respeecher.com/blog/respeecher-gives-voice-michael-york-healthcare-initiative`, 2021.

[3] Tomoki Toda, Alan W Black, and Keiichi Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *TASLP*, vol. 15, no. 8, 2007.

[4] Kou Tanaka, Hirokazu Kameoka, Takuhiro Kaneko, and Nobukatsu Hojo, "Atts2s-VC: Sequence-to-sequence voice conversion with attention and context preservation mechanisms," in *ICASSP*, 2019.

[5] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *ICME*, 2016.

[6] Wen-Chin Huang, Tomoki Hayashi, Shinji Watanabe, and Tomoki Toda, "The Sequence-to-Sequence Baseline for the Voice Conversion Challenge 2020: Cascading ASR and TTS," in *Interspeech BC/VCC workshop*, 2020.

[7] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *SLT*, 2018.

[8] Kaizhi Qian et al., "AutoVC: Zero-shot voice style transfer with only autoencoder loss," in *ICML*, 2019.

[9] Yi Zhao et al., "Voice conversion challenge 2020: Intra-lingual semi-parallel and cross-lingual voice conversion," in *Interspeech BC/VCC workshop*, 2020.

[10] Adam Polyak et al., "Speech resynthesis from discrete disentangled self-supervised representations," in *Interspeech*, 2021.

[11] Benjamin van Niekerk, Leanne Nortje, and Herman Kamper, "Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge," in *Interspeech*, 2020.

[12] Wen-Chin Huang, Yi-Chiao Wu, and Tomoki Hayashi, "Any-to-one sequence-to-sequence voice conversion using self-supervised discrete speech representations," in *ICASSP*, 2021.

[13] Wei-Ning Hsu et al., "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *arXiv preprint arXiv:2106.07447*, 2021.

[14] Jan van Gemert, Cor J. Veenman, Arnold W. M. Smeulders, and Jan-Mark Geusebroek, "Visual Word Ambiguity," *TPAMI*, vol. 32, no. 7, 2010.

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[16] Tu Anh Nguyen et al., "The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling," in *NeurIPS SAS Workshop*, 2020.

[17] Benjamin van Niekerk, Leanne Nortje, Matthew Baas, and Herman Kamper, "Analyzing speaker information in self-supervised models to improve zero-resource speech processing," in *Interspeech*, 2021.

[18] Shu wen Yang et al., "Superb: Speech processing universal performance benchmark," in *Interspeech*, 2021.

[19] Jonathan Shen et al., "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," in *ICASSP*, 2018.

[20] Julian Zaïdi, Hugo Seuté, Benjamin van Niekerk, and Marc-André Carbonneau, "Daft-exprt: Robust prosody transfer across speakers for expressive speech synthesis," *arXiv preprint arXiv:2108.02271*, 2021.

[21] Aäron van den Oord et al., "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[22] Jungil Kong et al., "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *NeurIPS*, 2020.

[23] Keith Ito and Linda Johnson, "The LJ speech dataset," `keithito.com/LJ-Speech-Dataset`, 2017.

[24] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[25] Kyubyong Park and Thomas Mulc, "CSS10: A collection of single speaker speech datasets for 10 languages," in *Interspeech*, 2019.

[26] Daniel van Niekerk et al., "Rapid development of TTS corpora for four South African languages," in *Interspeech*, 2017.

[27] Jacob Kahn et al., "Libri-Light: A benchmark for ASR with limited or no supervision," in *ICASSP*, 2020.

[28] Xinjian Li et al., "Universal phone recognition with a multilingual allophone system," in *ICASSP*, 2020.

[29] David R Mortensen, Siddharth Dalmia, and Patrick Littell, "Epitran: Precision G2P for many languages," in *LREC*, 2018.

[30] David Snyder et al., "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP*, 2018.

[31] Mark Cartwright, Bryan Pardo, Gautham J. Mysore, and Matthew Hoffman, "Fast and easy crowdsourced perceptual audio evaluation," in *ICASSP*, 2016.