A comparison of handcrafted, parameterized, and learnable features for speech separation

Wenbo Zhu, Mou Wang, Xiao-Lei Zhang, Susanto Rahardja
CIAIC, School of Marine Science and Technology, Northwestern Polytechnical University, China
E-mail: {wbzhu, wangmou21}@mail.nwpu.edu.cn, {xiaolei.zhang, susanto}@nwpu.edu.cn

***Abstract*—The design of acoustic features is important for speech separation. It can be roughly categorized into three classes: handcrafted, parameterized, and learnable features. Among them, learnable features, which are trained with separation networks jointly in an end-to-end fashion, become a new trend of modern speech separation research, e.g., convolutional time domain audio separation network (Conv-Tasnet), Dual-path RNN(DPRNN), Dual-path Transformer Network(DPTNet). At the same time, handcrafted and parameterized features are also shown competitively in very recent studies. However, a systematic comparison across the three kinds of acoustic features has not been conducted yet. This paper compares them in the framework of Conv-Tasnet, DPRNN, and DPTNet by setting its encoder and decoder with different acoustic features. We also generalize the handcrafted multi-phase gammatone filterbank (MPGTF) to a new parameterized multi-phase gammatone filterbank (ParaMPGTF). Experimental results on the WSJ0-2mix corpus and WHAMR! corpus show that if the decoder is learnable, then setting the encoder to STFT, MPGTF, ParaMPGTF, and learnable features lead to similar performance on Conv-Tasnet, DPRNN, and DPTNet; and when the pseudo-inverse transforms of STFT, MPGTF, and ParaMPGTF are used as the decoders on Conv-Tasnet, the proposed ParaMPGTF performs better than the other two handcrafted features.**

I. Introduction

Speech separation aims to separate a mixture of multiple speech sources into its components. In this paper, we study deep learning based speaker-independent speech separation, which does not require training and test speakers to be the same [1]. Hershey *et al.* first addressed the problem by deep clustering [2]. Since then, several methods have been proposed, such as permutation invariant training [3], [4] and deep attractor networks [5] which aim to estimate a time-frequency mask for each speaker. Among the methods, the magnitude spectrogram of short-time Fourier transform (STFT) is the most widely used acoustic feature. However, when recovering the time-domain speech from the separated magnitude spectrograms, the noisy phase has to be used, which results in suboptimal performance.

To remedy this weakness, learnable features, which learn a network for the transforms between the time-domain signal and its time-frequency spectrogram are becoming a new trend. Representative ones include one-dimensional convolution (1D-conv) filters [6]–[11]. Because the transforms are jointly trained with the separation network, and also because they do not need additional handcrafted operations, they lead to improved performance over STFT. Among the time-domain speech separation methods, convolutional time domain audio separation network (Conv-Tasnet), which reaches outstanding separation performance with a frame length of only 2 ms, received much attention.

Several recent work studied acoustic features with Conv-Tasnet. For example, Ditter and Gerkmann [12] used a handcrafted feature, named multi-phase gammatone filterbank (MPGTF), to replace the 1D-conv learnable feature of the encoder, which leads to improvement over the original Conv-Tasnet in terms of the scale-invariant source-to-noise ratio (SI-SNR). Pariente *et al.* [13] extended the parameterized filters introduced in [14] to complex-valued analytic filters, and then proposed a similar analytic extension for the 1D-conv filter as well. The analytic 1D-conv filter improves the performance as well. The aforementioned positive results demonstrate that handcrafted and parameterized features are also competitive to the state-of-the-art learnable features.

The above works mainly focus on the front-end features, and the separation network design is also essential. Yi Luo *et al.* [10] used a dual-path RNN to replace the temporal convolutional network (TCN) as a separator of the Conv-Tasnet, which significantly reduced model size and improve the SI-SNR. Based on DPRNN, Chen *et al.* [11] introduced an improved transformer [15] to replace the RNN of the DPRNN, and the dual-path structure was reserved, leading to superior separation performance.

However, there lacks a comparison between the handcrafted, parameterized, and learnable features. Motivated by replacing the encoder or decoder with handcrafted features, in this paper, we compared the three kinds of features in three frameworks of Conv-Tasnet, DPRNN, and DPTNet. To understand the connection between the three kinds of features, we proposed a parameterized extension of MPGTF, named parameterized MPGTF (ParaMPGTF). The center frequencies and bandwidths of ParaMPGTF are jointly trained with the separation network. We conducted an experimental comparison between STFT, MPGTF, ParaMPGTF, and learnable features on WSJ0-2mix [2] and WHAMR! [16]. Experimental results show that if the decoder is learnable, setting the encoder to any of the comparison features leads to similar performance on Conv-Tasnet, DPRNN, DPTNet. We have also compared STFT, MPGTF, and ParaMPGTF when their (pseudo) inverse transforms are used as the decoders on Conv-Tasnet. Results show that the proposed ParaMPGTF performs better than the other two handcrafted features.

This paper is organized as follows. Section II presents the

978-988-14768-9-0/21/$31.00 ©2021 APSIPA

comparison framework and the proposed ParaMPGTF. Section III presents the experimental results. Finally, we conclude our findings in Section IV.

II. Methods

A. Preliminary

Given C speech sources $\{\mathbf{s}_c(t)\}_{c=1}^{C}$ with t as the index of time samples, their mixed signal is

$$\mathbf{x}(t) = \sum_{c=1}^{C} \mathbf{s}_c(t) \tag{1}$$

The problem of speech separation can be described as producing an accurate estimate $\hat{\mathbf{s}}_c(t)$ for $\mathbf{s}_c(t)$ from $\mathbf{x}(t)$.

The framework in this study is Conv-Tasnet [7], DPRNN [10], and DPTNet [11]. These three frameworks have the same structure, which consists of three main parts—an encoder, a separation network, and a decoder. These use a small frame size in the encoder and decoder to reduce the time delay significantly. The encoder and decoder are learnable 1D-conv filters, which perform like transform between the time-domain signal and time-frequency features. The separation network of Conv-Tasnet is a fully-convolutional separation module that consists of stacked one-dimensional dilated convolutional blocks [17], [18]. DPRNN's separation network consists of three stages: segmentation, block processing, and overlap-add. The core stage, block processing, passed the tensor generated by the segmentation stage to stacked DPRNN blocks to apply local iteratively (intra-chunk) and global (inter-chunk) modeling alternately. The output is transformed back to a sequential output with the overlap-add method. The separation network of DPTNet had replaced the RNN in DPRNN blocks with an improved transformer while retaining the dual-path structure in DPRNN. These frameworks are optimized with the scale-invariant source-to-noise ratio (SI-SNR) loss [5].

B. Comparison framework

The comparison uses handcrafted transforms, parameterized transforms, and learnable filters as the encoder and decoder. The encoder can be thought of as s set of N filters of length L. The output of the encoder is a time-frequency representation produced from the convolution of the mixed speech input signal with the filter:

$$\mathbf{X}(n,i) = \mathcal{H}(\sum_{l=0}^{L-1} \mathbf{x}(iD+l)\mathbf{h}_n^{\mathrm{Enc}}(L-l)) \tag{2}$$

where n is the filter index, i is the frame index, D is the frame shift, $\mathbf{h}_n^{Enc}(\cdot)$ is the n-th filter of the filterbanks, l denotes the sample index in a frame, and $\mathcal{H}(\cdot)$ is the rectified linear unit (ReLU) to ensure that the representation is non-negative. In the comparison, $\mathbf{h}_n^{\mathrm{Enc}}(\cdot)$ can be any of the three kinds of feature transforms.

The decoder reconstructs the time-domain signal of the c-th speaker $\hat{\mathbf{s}}_c \in \mathbb{R}^T$. The output of the decoder is:

$$\hat{\mathbf{s}}_c(k,i) = \sum_{n=0}^{N-1} \hat{\mathbf{S}}_c(n,i)\mathbf{h}_{N-n}^{\mathrm{Dec}}(k) \tag{3}$$

where $\hat{\mathbf{S}}_c(n,i)$ is the output of the separation network for the c-th speaker, k is the index of the filter weight, $\mathbf{h}_n^{\mathrm{Dec}}(\cdot)$ is the n-th filter of the decoder, and $\hat{\mathbf{s}}_c(k,i)$ is the estimate of the c-th speech source at the i-th frame. To decode the frame-shift operation between speech frames, the decoder further calculates $\hat{\mathbf{s}}_c(t) = \sum_{i=-\infty}^{\infty} \hat{\mathbf{s}}_c(t-iD,i)$.

The comparison uses STFT, MPGTF, ParaMPGTF, and learnable filters as $\mathbf{h}_n^{\mathrm{Enc}}(\cdot)$ with their inverse transforms as $\mathbf{h}_{N-n}^{\mathrm{Dec}}(\cdot)$, where the proposed ParaMPGTF is presented in the next subsection.

C. Parameterized multi-phase gammatone filterbank

Gammatone filterbank, which mimics the masking effect of the human auditory system, are good features for speech separation [19]. The impulse response function $\gamma(t)$ of a gammatone filter is

$$\gamma(t) = \alpha t^{n-1}\exp(-2\pi bt)\cos(2\pi f_c t + \phi) \tag{4}$$

where n is the order, b is a bandwidth parameter, f_c is the centre frequency of the filter, $t>0$ is the time in seconds, α is the amplitude, and ϕ is the phase shift. Ditter and Gerkmann [12] extended the classical gammatone filterbank to MPGTF. The extension has the following three three aspects: First, the length of the filters is set to 2ms, which keeps the system a low latency. Second, for each filter $h_n^{\mathrm{Enc}}(\cdot)$, MPGTF introduces $-h_n^{\mathrm{Enc}}(\cdot)$ to ensure that, for each centre frequency, at least one filter contains energy. Third, the phase shift ϕ varies at the same centre frequency. The details of MPGTF can be found in [12].

From (4), we observe that the bandwidth parameter b and filter centre frequency f_c are two important parameters. They are determined by the equivalent rectangular bandwidth (ERB) [20] using a rectangular band-pass filter:

$$\mathrm{ERB}(f_c, c_1, c_2) = c_1 + \frac{f_c}{c_2} \tag{5}$$

$$f_c = c_2(\mathrm{ERB} - c_1) \tag{6}$$

$$b = \frac{\mathrm{ERB}\sqrt{(n-1)!}}{\pi\left((2n-2)!\right)2^{2-2n}} \tag{7}$$

where c_1 and c_2 are two parameters. Traditionally, the parameters c_1 and c_2 are set to 24.7 and 9.265 respectively in experience [20]. This empirical setting may not be accurate enough, which may lead to suboptimal performance.

To overcome this issue, we propose ParaMPGTF which trains the filterbank parameters c_1 and c_2 in MPGTF jointly with the network. For each iteration, we update the parameter b by (7) and the centre frequencies $f_{c_1}, f_{c_2}, \ldots, f_{c_M}$ by:

$$f_{c_j} = \mathrm{ERB}_{\mathrm{scale}}^{-1}(\mathrm{ERB}_{\mathrm{scale}}(f_{c_{j-1}}) + 1) \tag{8}$$

according to the updated c_1 and c_2, where f_{c_j} denotes the centre frequency of the j-th filter, M is the number of filters in the filterbank, $\mathrm{ERB}_{\mathrm{scale}}$ denotes the ERB scale calculated by integrating $1/\mathrm{ERB}(f_c)$ across frequency, and $\mathrm{ERB}_{\mathrm{scale}}^{-1}$ is the inverse of $\mathrm{ERB}_{\mathrm{scale}}$. In practice, $\mathrm{ERB}_{\mathrm{scale}}$ and $\mathrm{ERB}_{\mathrm{scale}}^{-1}$

are calculated by:

$$\mathrm{ERB}_{\mathrm{scale}}(f_{\mathrm{Hz}}) = c_2 \log(1 + \frac{f_{\mathrm{Hz}}}{c_1 c_2}) \quad (9)$$

$$\mathrm{ERB}^{-1}_{\mathrm{scale}}(\mathrm{ERB}_{\mathrm{scale}}) = c_1 c_2 (\mathrm{e}^{\frac{\mathrm{ERB}_{\mathrm{scale}}}{c_2}} - 1) \quad (10)$$

where f_{Hz} denotes a frequency variable. After obtaining $f_{c_1}, \ldots, f_{c_M}$ and b, we obtain the updated filterbank according to (4). To make ParaMPGTF a meaningful filterbank, $f_{c_1}, f_{c_2}, \ldots, f_{c_M}$ should be constrained between 100 Hz to 4000 Hz. To satisfy this constraint, we fix f_{c_1} to 100 Hz in the entire training process.

To summarize, ParaMPGTF combines the data-driven scheme with MPGTF [12]. It inherits the changes of MPGTF.

III. EXPERIMENTALS AND RESULTS

A. Dataset

We conducted the comparison on two-speaker speech separation using the WSJ0-2mix dataset [2] and the WHAMR! dataset [16]. WSJ0-2mix contains 30 hours training data, 10 hours development data, and 5 hours test data. The mixtures in WSJ0-2mix were generated by first randomly selecting different speakers and utterances in the Wall Street Journal (WSJ0) training set si_tr_s, and then mixing them at a random signal-to-noise ratio (SNR) level between -5 dB and 5 dB [7]. The utterances in the test set were from 16 unseen speakers in the si_dt_05 and si_et_05 directories of the WSJ0 dataset. All waveforms were resampled to 8 kHz. The WHAMR! dataset is an augmented version of WSJ0-2mix with synthetic reverberated sources and noise. There are four configurations: a min condition where the mixture is trimmed to the length of the shorter utterance and the corresponding non-trimmed max condition, both available at 8 kHz and 16 kHz sampling rate. Here we use the min condition with 8 kHz sample rate configuration.

B. Experimental setup

The network was trained for 200 epochs on 4-second long segments. Adam was used as the optimizer with an initial learning rate of 0.001. The learning rate was halved if the performance of development set was not improved in 5 consecutive epochs. The network training procedure was early stopped when the performance on the development set has not been improved within the last 10 epochs. The hyperparameters of the network followed the setting in [12], where the number of filters N is 512. If not explicitly stated in the paper, the mask functions of the separation network are rectified linear unit(ReLU).

For ParaMPGTF, we set the order n and amplitude α to 2 and 1 respectively. We initialize c_1 and c_2 to their empirical values, i.e. $c_1 = 24.7$ and $c_2 = 9.265$.

We used SI-SNR as the evaluation metric [5]. We reported the average results over all 3000 test mixtures.

TABLE I
COMPARISON OF DIFFERENT ENCODERS WHEN THE DECODERS ARE SET TO LEARNABLE FILTERS.

Encoder	Decoder	Mask activation	SI-SNR (dB)	
			Dev	Test
Learned	Learned	Sigmoid	17.61	16.92
Learned	Learned	ReLU	17.45	16.89
MPGTF	Learned	ReLU	17.66	17.20
ParaMPGTF	Learned	ReLU	17.71	17.06
STFT	Learned	ReLU	**17.96**	**17.28**

TABLE II
COMPARISON OF c_1 AND c_2 BETWEEN MPGTF AND PARAMPGTF WHEN THE DECODERS ARE SET TO LEARNABLE FEATURES.

	MPGTF	ParaMPGTF
c_1	24.7	25.09
c_2	9.265	9.198

C. Results with learnable decoders of Conv-Tasnet on WSJ0-2mix dataset

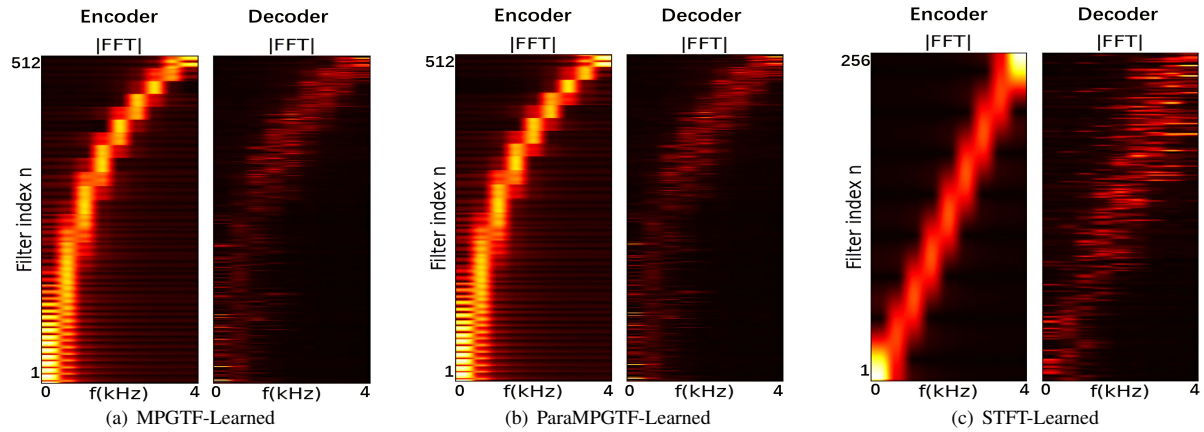
We first conducted a comparison between STFT, MPGTF, ParaMPGTF, and learnable features when the decoders were set to the learnable features. The comparison results are listed in Table I. From the table, we observe that the four features do not yield fundamentally different performance. If we look at the details, we find that STFT reaches the highest SI-SNR in both the development set and the test set. MPGTF and ParaMPGTF show competitive performance, where ParaMPGTF performs slightly better than MPGTF on the development set, and slightly worse than the latter on the test set.

Fig. 1 shows the magnitude spectrograms of the MPGTF, ParaMPGTF, and STFT encoders with their corresponding learnable decoders, where we only plot the STFT bins with indices from 1 to 256 [21], [22] since that the real and imaginary parts share similar patterns. The filters are uniformly distributed in the frequency range from 0 Hz to 4000 Hz. From the figure, we see that the magnitude spectrograms of ParaMPGTF and MPGTF are similar. This phenomenon not only accounts for their similar performance, but also demonstrates that the parameterized feature is able to be optimized successfully. As a byproduct, it shows that (i) MPGTF is a well-designed handcrafted feature; (ii) the learnable decoders are able to learn effective inverse transforms of their encoders.

Table II lists the comparison between the handcrafted c_1 and c_2 in MPGTF and the optimized c_1 and c_2 in ParaMPGTF. From the table, we see that the two groups of the parameters are similar, which further accounts for the similar performance of MPGTF and ParaMPGTF.

D. Results with (pseudo) inverse transform decoders of Conv-Tasnet on WSJ0-2mix dataset

In this experiment, we set the encoder to STFT, MPGTF, and ParaMPGTF respectively, and set the decoder to their inverse transforms accordingly.



(a) MPGTF-Learned (b) ParaMPGTF-Learned (c) STFT-Learned

Fig. 1. Visualization of different configuration of encoder and learned decoder magnitudes of their FFTs. The left is a MPGTF-based encoder, the middle is a ParaMPGTF-based encoder and the right is a STFT-based encoder.

TABLE III
COMPARISON OF ENCODERS AND DECODERS WITH DIFFERENT FEATURES. THE MASK ACTIVATION FUNCITON IS RELU.

Encoder	**Decoder**	SI-SNR (dB)	
		Dev	Test
MPGTF	MPGTF Pseudo Inv.	16.32	15.73
ParaMPGTF	ParaMPGTF Pseudo Inv.	**16.64**	**16.04**
STFT	ISTFT	16.31	15.82

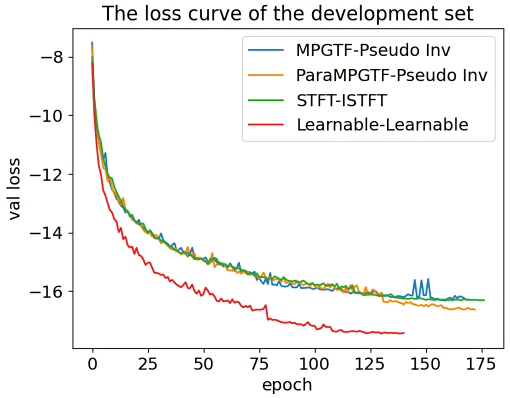


Fig. 2. Convergence curves of different encoder-decoder pairs in the training process.

Table III lists the performance of MPGTF, ParaMPGTF, and STFT with their (pseudo) inverse transforms. From the table, we see that the performance of the three comparison methods is similar in general. If we look into the details, we see that the proposed ParaMPGTF reaches the best performance among the comparison methods on both the development set and the test set, which demonstrates the potential of the parameterized training strategy in improving conventional handcrafted features.

Fig. 2 shows the convergence curves of the deep models on the development set when the decoders are set to the (pseudo) inverse transforms of their encoders. From the figure, we find that the learnable feature converges faster than the handcrafted and parameterized features. Although the handcrafted features and ParaMPGTF converge in a similar rate at the early training stage, ParaMPGTF converges faster at the late training stage.

E. Results with learnable decoders of Conv-Tasnet, DPRNN, DPTNet on WHAMR! dataset

The above experimental results are obtained under ideal conditions. In this experiment, we set the decoder to the learnable features, and we compare four different features of Conv-Tasnet, DPRNN, and DPTNet on a simulated real condition. The comparison results of Conv-Tasnet based are list in Table IV. In this table, the four tasks: sep clean, sep noisy, sep reverb, and sep noisy and reverb, which means we separate speech under what conditions, e.g., sep noisy and reverb task mean we separate speech in an environment where exists noise and reverberation. From the table, we observe that the four features generally have a similar separation performance on the four tasks. If we look at the detail, our proposed ParaMPGTF works better than MPGTF when reverberation exists. Except on the sep reverb task, STFT reaches the best performance on others tasks, which slightly better than the learnable feature, MPGTF, and ParaMPGTF.

Table V and table VI present the performance on the sep clean and sep noisy and reverb tasks of DPRNN and DPTNet, respectively. From the table V we observe that the learnable feature performs better than the other three features on the sep clean tasks, and on the sep noisy and reverb task, the STFT feature has a slight improvement than the learnable feature, MPGTF, and ParaMPGTF. By the way, our proposed ParaMPGTF performs better than MPGTF in both conditions. When using the DPTNet, we find that the learnable feature surpasses other features in terms of SI-SNR on both tasks, our proposed ParaMPGTF slightly better than MPGTF on sep clean task and slightly worse than the latter on sep noisy and reverb task.

TABLE IV
COMPARISON OF DIFFERENT ENCODERS WHEN THE DECODERS ARE SET TO LEARNABLE FILTERS ON WHAMR! WITH THE DIFFERENT TASKS ON THE CONV-TASNET. THE MASK ACTIVATION FUNCTION IS RELU.

Task	Noisy	Reverb	Encoder/Decoder	SI-SNR (dB)
sep clean	×	×	Learned/Learned	16.89
	×	×	STFT/Learned	**17.28**
	×	×	MPGTF/Learned	17.20
	×	×	MPGTFPara/Learned	17.06
sep noisy	√	×	Learned/Learned	9.34
	√	×	STFT/Learned	**9.35**
	√	×	MPGTF/Learned	9.24
	√	×	MPGTFPara/Learned	9.19
sep reverb	×	√	Learned/Learned	7.10
	×	√	STFT/Learned	6.78
	×	√	MPGTF/Learned	7.07
	×	√	MPGTFPara/Learned	**7.12**
sep noisy and reverb	√	√	Learned/Learned	4.65
	√	√	STFT/Learned	**4.68**
	√	√	MPGTF/Learned	4.55
	√	√	MPGTFPara/Learned	4.63

TABLE V
COMPARISON OF DIFFERENT ENCODERS WHEN THE DECODERS ARE SET TO LEARNABLE FILTERS ON WHAMR! WITH THE DIFFERENT TASKS ON THE DPRNN. THE MASK ACTIVATION FUNCTION IS RELU.

Task	Noisy	Reverb	Encoder/Decoder	SI-SNR (dB)
sep clean	×	×	Learned/Learned	**17.44**
	×	×	STFT/Learned	16.99
	×	×	MPGTF/Learned	16.60
	×	×	MPGTFPara/Learned	16.82
sep noisy and reverb	√	√	Learned/Learned	4.13
	√	√	STFT/Learned	**4.40**
	√	√	MPGTF/Learned	4.16
	√	√	MPGTFPara/Learned	4.29

TABLE VI
COMPARISON OF DIFFERENT ENCODERS WHEN THE DECODERS ARE SET TO LEARNABLE FILTERS ON WHAMR! WITH THE DIFFERENT TASKS ON THE DPTNET. THE MASK ACTIVATION FUNCTION IS RELU.

Task	Noisy	Reverb	Encoder/Decoder	SI-SNR (dB)
sep clean	×	×	Learned/Learned	**16.43**
	×	×	STFT/Learned	15.90
	×	×	MPGTF/Learned	15.04
	×	×	MPGTFPara/Learned	15.30
sep noisy and reverb	√	√	Learned/Learned	**4.70**
	√	√	STFT/Learned	4.67
	√	√	MPGTF/Learned	4.33
	√	√	MPGTFPara/Learned	4.20

IV. CONCLUSIONS

In this paper, we have proposed a parameterized multi-phase gammatone filterbank. ParaMPGTF jointly learns the core parameters of MPGTF with the separation network. We have also compared handcrafted, parameterized, and learnable features in the same experimental framework, which is to our knowledge the first time that the three kinds of features are compared together, where the features in comparison are STFT, MPGTF, ParaMPGTF, and learnable features. Experiment results show that when the decoders are set to learnable features, the four features behave similarly whenever the separation network is Conv-Tasnet, DPRNN, or DPTNet. STFT behave slightly better than the others in most condition. When the decoders are set to the (pseudo) inverse transforms of the encoders on the Conv-Tasnet, ParaMPGTF performs better than the handcrafted features.

REFERENCES

[1] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[2] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 31–35.

[3] D. Yu, M. Kolbk, Z. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 241–245.

[4] M. Kolbk, D. Yu, Z. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[5] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 246–250.

[6] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," 2017.

[7] ——, "Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation," 2018.

[8] A. Pandey and D. Wang, "Tcnn: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6875–6879.

[9] Z. Shi, H. Lin, L. Liu, R. Liu, J. Han, and A. Shi, "Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks," 2019.

[10] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 46–50.

[11] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," *arXiv preprint arXiv:2007.13975*, 2020.

[12] D. Ditter and T. Gerkmann, "A multi-phase gammatone filterbank for speech separation via tasnet," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 36–40.

[13] M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "Filterbank design for end-to-end speech separation," 2019.

[14] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," 2018.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[16] M. Maciejewski, G. Wichern, E. McQuinn, and J. Le Roux, "Whamr!: Noisy and reverberant single-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 696–700.

[17] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, pp. 47–54.

[18] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1003–1012.

[19] R. D. Patterson, K. Robinson, J. Holdsworth, D. Mckeown, C. Zhang, and M. Allerhand, "Complex sounds and auditory images," *Auditory Physiology and Perception*, pp. 429–446, 1992.

[20] V. Hohmann, "Frequency analysis and synthesis using a gammatone filterbank," *Acta Acustica United with Acustica*, vol. 88, no. 3, pp. 433–442, 2002.

[21] A. Pandey and D. Wang, "A new framework for cnn-based speech enhancement in the time domain," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 7, pp. 1179–1188, 2019.

[22] A. Pandey and D. Wang, "A new framework for cnn-based speech enhancement in the time domain," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 7, pp. 1179–1188, 2019.