Time-Domain Speech Separation Networks With Graph Encoding Auxiliary

Tingting Wang, Zexu Pan, *Member, IEEE*, Meng Ge, Zhen Yang, *Senior Member, IEEE*, and Haizhou Li, *Fellow, IEEE*

***Abstract*—End-to-end time-domain speech separation with masking strategy has shown its performance advantage, where a 1-D convolutional layer is used as the speech encoder to encode a sliding window of waveform to a latent feature representation, i.e. an embedding vector. A large window leads to low resolution in the speech processing, on the other hand, a small window offers high resolution but at the expense of high computational cost. In this work, we propose a graph encoding technique to model the fine structural knowledge of speech samples in a window of reasonable size. Specifically, we build a graph representation for each latent representation, and encode the structural details with a graph convolutional network encoder. The encoded graph feature representation complements the original latent feature representation and benefits the separation and reconstruction of speech. Experiments on various models and datasets show that our proposed encoding technique significantly improves the speech quality over other time-domain speech encoders.**

***Index Terms*—Source separation, single channel, deep learning, graph signal processing, graph neural networks.**

I. Introduction

Speech separation seeks to disentangle a multi-talk speech mixture into individual speakers in a *cocktail-party* scenario [1], [2]. It has attracted extraordinary attention in recent years, in a quest for an indispensable speech acquisition front-end for many downstream tasks like speech recognition [3], speaker localization [4], active speaker detection [5], and speech emotion recognition [6].

The Conv-TasNet [7] is one of the successful implementations of time-domain speaker separation network. It employs a 1-D convolutional layer to replace the the short-time Fourier transform (STFT) operation as a speech encoder to generate a learned latent feature representation, which avoids the phase reconstruction problem [7], [8]. Since then, the time-domain implementations have attracted increasing attention due to their superior performance [9], [10], [11], [12], [13].

Among the time-domain implementations, the study on the Dual-Path-RNN (DPRNN) shows that, by reducing the window size, we may obtain a fine-grained waveform encoding, which improves the separation performance [14] at high computational cost. In another line of research, hand-crafted speech encoders are incorporated with the learnable ones to improve the interpretability of feature representation. In [15], [16], a deterministic gammatone filterbank was proposed to replace the learned encoder, so as to reduce the model variance thus avoid overfitting. In this work, we take a different approach, that is, to study how time-domain speech separation technique captures high resolution details (dependency, similarity, smoothness) of the waveform with a medium size encoding window [17], [18].

A graph is a compact, efficient, and scalable representation of data and its structural relationships [19], that graph signal processing (GSP) [20] is based on. Following the graph convolution definition [21], the graph convolution network (GCN) [22] extends the convolutional neural network to deal with signals on a graph. It has been the prevalent model in structural relationship learning and is applied to various problems, such as link prediction [23], natural language processing [24] and video representation learning [25]. Moreover, in speech processing tasks, Panagiotis et al. in [26] viewed different channels as vertexes of the graph for multi-channel speech enhancement. Amir et al. in [27] modeled speech signals as a cycle/line graph for speech emotion recognition.

A time-domain speech separation network usually consists of a speech encoder, a mask estimator, and a speech decoder. A small speech encoding window usually leads to small frame shift, thus high frame rate and computational cost. Inspired by the success of GCN on speech enhancement and other speech graph modeling [28], [29], we propose a graph encoder to

Manuscript received 2 December 2022; revised 3 February 2023; accepted 4 February 2023. Date of publication 9 February 2023; date of current version 22 February 2023. This work was supported in part by the National Natural Science Foundation of China under Grants 62071242 and 62271432, in part by Internal Project Fund Shenzhen Research Institute of Big Data under Grant T00120220002, in part by the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen under Grant B10120210117-KP02, in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX19-0897, and in part by DFG (German Research Foundation) through Germany's Excellence Strategy (University Allowance, EXC 2077, University of Bremen). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joseph Keshet. *(Corresponding author: Meng Ge.)*

Zexu Pan is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: pan_zexu@u.nus.edu).

Meng Ge is with the Tianjin Key Laboratory of Cognitive Computing and Application, College of Intelligence and Computing, Tianjin University, Tianjin 300072, China (e-mail: gemeng@tju.edu.cn).

Tingting Wang and Zhen Yang are with the Department of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210023, China (e-mail: 2018010215@njupt.edu.cn; yangz@njupt.edu.cn).

Haizhou Li is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077, with the Shenzhen Research Institute of Big Data, School of Data Science, The Chinese University of Hong Kong, Shenzhen 518172, China, with the University of Bremen, 28359 Bremen, Germany, with the Kriston AI, Xiamen 361021, China, and also with the National University of Singapore, Singapore 119077 (e-mail: haizhou.li@nus.edu.sg).

Digital Object Identifier 10.1109/LSP.2023.3243764

augment the learnable speech encoder in this letter. We take the advantage of the graphs and GCN to capture high-resolution details of the waveform. The graph encoder enables the learning of the high-order contextual knowledge of speech in each sliding window without increasing the frame rate.

The learnable speech encoder takes a time-domain window of signal as input and generates an embedding vector. We consider the elements in the embedding vector as the vertexes in a graph. We build a graph representation by focusing on the presence or absence of structural details between the vertexes. Then we construct a two-layer GCN to encode the high-order structural details between the vertexes. Finally, the embedding extracted by the graph encoder and the speech encoder are fused for the mask estimation. We implement our graph encoding technique on popular architectures namely Conv-TasNet and DPRNN, the experiments show that the proposed graph encoder improves the single-channel speech separation performance on various datasets, i.e., WSJ0-2Mix [8], WHAM! [30] and Libri2Mix [31].

II. Related Work

A. Graph Signal Processing

Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ denote as a digraph with a set of vertices $\mathcal{V}$ and a set of edge links $\mathcal{E}$, such that if vertex i is connected to vertex j, then $(i, j) \in \mathcal{E}$ [32]. Considering that the focus of the letter is on investigating digraph signals, we define the non-symmetric adjacency matrix $\mathbf{A}$ of as a sparse matrix with nonzero elements $\mathbf{A}(i, j)$ if and only if $(i, j) \in \mathcal{E}$; and the value of $\mathbf{A}(i, j)$ reflects the strength of the connection from i to j.

Given a graph G, following the description of GCN in [22], a multi-layer GCN based on the layer-wise propagation rule for signals $\mathbf{S} \in \mathbb{R}^{|\mathcal{V}| \times M}$ with M-dimension features takes the form

$$\mathbf{H}^{l+1} = \sigma((\mathbf{D})^{-1/2}\mathbf{A}(\mathbf{D})^{-1/2}\mathbf{H}^{l}\boldsymbol{W}^{(l)}), \tag{1}$$

where $\mathbf{H}^{l} \in \mathbb{R}^{|\mathcal{V}| \times M}$ is the l^{th} layer output and $\mathbf{H}^{0} = \mathbf{S}$. $\mathbf{D}$ is a diagonal matrix of which the diagonal element is the sum of each row entry of $\mathbf{A}$, $\boldsymbol{W}^{(l)}$ is a layer-specific trainable weight matrix, and $\sigma(\cdot)$ represents an activation function. More of GCN can be found in [22].

B. Adjacency Matrix of Speech Graph Signals

In our previous work [28], [29], we study how to infer a suitable graph representation for speech signals. To be specific, we view each discrete speech sample as a vertex; the sample value is viewed as a signal residing on the vertex. In this way, a speech signal is processed as a graph signal residing on a directed graph. The one-to-one mapping between the τ_{th} speech sample $x(\tau)$ and the value of the τ_{th} vertex v_{τ} satisfies

$$s_g : \mathbb{R} \rightarrow \mathcal{V}, x(\tau) \rightarrow v_{\tau}, \tag{2}$$

where $\mathbb{R}$ and $\mathcal{V}_s$ represent the real number set and a set of vertices of speech graph signals (SGSs), respectively. We formulate a methodology to derive a graph model for speech signals from the linear shift-invariance graph filters, namely, the k-shift graph operator $\mathbf{A}_k$ [28]. Considering the causality and correlation among speech samples rather than their strength, $\mathbf{A}_k$ is defined as an adjacency matrix for speech graph signals, where k represents the graph shift numbers of speech signals. In this way, speech samples are mapped into the graph domain and constructed as SGSs residing on a graph G,

$$G = (\mathcal{V}, \mathcal{E}, \mathbf{A}_k) \tag{3}$$

III. Methodology

We propose a graph encoder that can be incorporated into the speech encoder of any time-domain speech separation networks, as illustrated in Fig. 1. In this way, the structural knowledge between the elements in the embedding vector can be encoded by the graph structure. We adopt the commonly used time-domain encoder, as in Conv-TasNet [7] and DPRNN [14]) to formulate the graph encoder.

A. Time-Domain Speech Encoder

The speech encoder encodes the discrete waveform $x(\tau)$ into a spectrum-like speech embedding sequence $\mathbf{X}(t)$. It consists of a 1D convolution layer $Conv1D$ with channel size N, window size L and stride $L/2$, followed by a rectified linear activation $ReLU$:

$$\mathbf{X}(t) = ReLU(Conv1D(x(\tau), N, L, L/2)). \tag{4}$$

where $t \in \{1, \ldots, T\}$, and T is the total number of latent features. The encoder module seeks to transform a window of the mixture waveform into a latent representation $\mathbf{X}(t)$.

B. Graph Encoder

It is shown that the structural relationship of the time-frequency bins can be modeled by graphical neural networks [26], [27] in frequency domain networks, but it is not easily implemented in time-domain networks. In Fig. 5 of Conv-TasNet [7], it was shown that the basis functions of the learned speech encoder resemble the property of the STFT operation, which serves as a frequency analyzer, and the majority of the basis functions concentrate at low frequency. In other words, the latent representation $\mathbf{X}(t)$ resembles the property of a spectrogram, and $\mathbf{X}(t)$ can be characterized by some graphical relations. We are motivated to capture such graphical relations of the elements in an embedding frame of a waveform window with a graph encoder.

We aim to build a graph representation $G_{(t)} = (\mathcal{V}_t, \mathbf{A}_k, \mathbf{A}_k)$ for every embedding $\mathbf{X}(t) \in \mathbb{R}^{N}$ here, and $\mathcal{V}_t$ is the vertex set. Under the SGSs framework [28], [29], we view each element in the embedding as a graph vertex and its value as the signals residing on the vertex, the number of vertex $|\mathcal{V}_t| = N$ here. We map the embedding $\mathbf{X}(t)$ into the graph domain and investigate the structural details among the elements. Assuming that if there is a structural detail between the i_{th} element indexed by the vertex v_i and the j_{th} element indexed by the vertex v_j, $\mathbf{A}_k(i, j)$ of $\mathbf{A}_k$ is set to be 1, otherwise $\mathbf{A}_k(i, j) = 0$. Here we consider that there exist the structural details among k feature points, where k is a hyper-parameter that needs to be defined.

We are particularly interested in the presence or absence of structural details among different embedding elements in this work, rather than the strength of their structural details, so we set the edge set $\mathcal{E}_k$ to be numerically equal to the graph adjacency

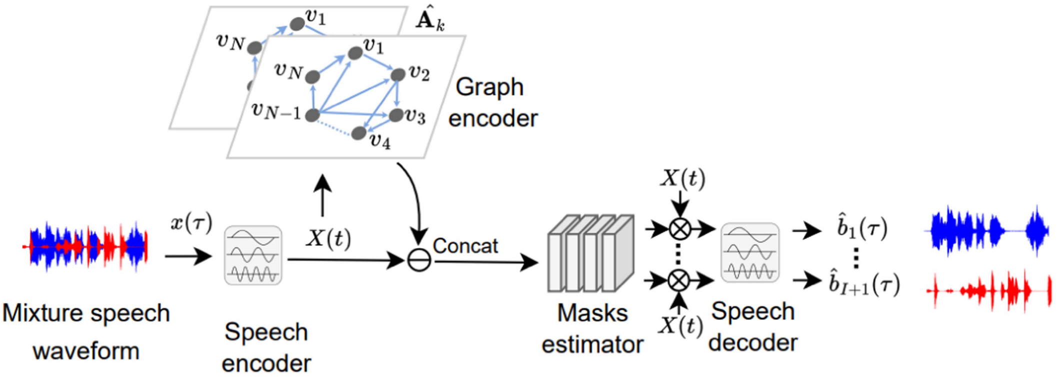

Fig. 1. The diagram of a typical time-domain speech separation system [7], [14], which consists of a speech encoder, a mask estimator, and a speech decoder. We propose a graph encoder, which can be incorporated into any time-domain speech encoder in a speech separation system.

matrix $\mathbf{A}_k$. Here $\mathbf{A}_k$ in the case of $k = 5$ is given as examples, that is,

$$\mathbf{A}_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & \cdots & 0 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & 1 & 1 & 0 \end{bmatrix} \tag{5}$$

Determining the graph representation of embedding vectors, for the embedding vector $\mathbf{X} \in \mathbb{R}^{N \times T}$, we first calculate $\widehat{\mathbf{A}}_k = (\mathbf{D}_k)^{-1/2} \mathbf{A}_k (\mathbf{D}_k)^{-1/2}$ in a pre-processing step, where $\mathbf{D}_k = \sum_j \widehat{\mathrm{A}}_k(i, j)$ is the diagonal matrix. Following (1), by applying $\mathbf{H}^0 = \mathbf{X}$ and $ReLU$ function, our two layer GCN-based $\widehat{\mathbf{A}}_k$ is represented as

$$\boldsymbol{\mathcal{F}} = ReLU(\widehat{\mathbf{A}}_k (ReLU((\widehat{\mathbf{A}}_k (\mathbf{X}^{\mathrm{T}} \boldsymbol{W}^{(0)})^{\mathrm{T}}))^{\mathrm{T}} \boldsymbol{W}^{(1)})^{\mathrm{T}}), \tag{6}$$

where $\boldsymbol{W}^{(0)} \in \mathbb{R}^{N \times N}$ represents an input-to-hidden weight matrix and $\boldsymbol{W}^{(1)} \in \mathbb{R}^{N \times N}$ is a hidden-to-output weight. Obtaining $\boldsymbol{\mathcal{F}} \in \mathbb{R}^{N \times T}$, we build up a new latent representation by concatenating $\mathbf{X}$ and $\boldsymbol{\mathcal{F}}$ and then transform it back to $\mathbf{X}$'s dimensions by a 1-D convolution operation with channel size $2N$, window size N and stride 1, followed by a rectified linear activation $ReLU$. Hence, a new mixture weight matrix $\mathbf{U} \in \mathbb{R}^{N \times T}$ is donated as

$$\mathbf{U} = ReLU(Conv1D(Concat(\mathbf{X}, \boldsymbol{\mathcal{F}}), N, 1)). \tag{7}$$

IV. Experimental Setup

A. Neural Architecture

The Conv-TasNet [7], DPRNN [14] are two popular neural architecture for time-domain speech separation. We therefore re-implement the two networks as our baseline. Unless mentioned otherwise, we set $L = 20$, $N = 256$, $B = 256$, $H = 512$, $P = 3$, $X = 3$ and $R = 4$ for Conv-TasNet, where B is the channel number in bottleneck and the residual paths' 1×1-conv blocks, H is the channel number in convolutional blocks, P is the kernel size in convolutional blocks, X is the number of convolutional blocks in each repeat and R is the repeat number. For the DPRNN network, we set $L = 20$, $N = 256$, $B = 64$, $R = 4$, and the chunk size $K = 80$ [14].

In the experiments, we incorporate the graph encoder into the Conv-TasNet to become GE-Conv-TasNet, as well as the DPRNN to become GE-DPRNN.

B. Datasets

We report the experiments on three datasets, namely WSJ0-2Mix [8], WHAM! [30] and Libri2Mix [31], that are all at sampled at 8 kHz.

WSJ0-2Mix is generated by randomly selecting utterances from different speakers in the Wall Street Journal (WSJ0) dataset, and mixing them at a random signal-to-noise ratio (SNR) from -5 dB to 5 dB. Ten hours of validation data are generated from speakers in si-tr-s. Five hours of evaluation data are generated in the same way from 16 unseen speakers in si-dt-05 and si-et-05.

The WHAM! dataset is generated by adding noisy samples to the WSJ0-2Mix at a SNR from -3 dB and 6 dB with respect to the loudest speaker.

The Libri2Mix dataset is generated by randomly selecting two speakers from the train-100 set in the Librispeech dataset [33] and mixing them at various SNRs uniformly sampled between 0 dB and 5 dB. The validation and test set are similarly created from unseen speakers in the Librispeech validation and test set [34].

C. Training

We adopt the scale-invariant source-to-distortion ratio (SI-SNR) as our training objective [7],

$$SI\text{-}SNR = 20 \log_{10} \frac{||\frac{<\hat{s},s>s}{||s||^2}||}{||\hat{s} - \frac{<\hat{s},s>s}{||s||^2}||}, \tag{8}$$

where $\hat{s}$ and s represent the estimated and ground truth clean signals respectively. During training, utterance-level permutation invariant training (uPIT) [35] is utilized to solve the permutation problem.

All networks are trained for 100 epochs on 5-second speech segments. We train the networks with the augmented graph encoder together from scratch. The learning rate was initialized to $1e^{-3}$ and halved if the validation loss is not improved in 3 consecutive epochs. Adam is used as our optimizer.

TABLE I
SI-SNRi (dB) of GE-Conv-TasNet on Validation Set of WSJ0-2Mix for Various k

Method	$k = 1$	$k = 10$	$k = 20$	$k = 30$
GE-Conv-TasNet (Ours)	16.88	16.76	17.01	16.53

TABLE II
SI-SNRi (dB) and MACs (G/s) With Different Window Size on Test Set of WSJ0-2Mix

Method	L	SI-SNRi	MACs
Conv-TasNet	10	15.63	55.10M
	20	15.01	27.78M
	40	14.95	13.91M
GE-Conv-TasNet (Ours)	20	16.28	28.20M
	40	15.48	14.12M

V. Results

We evaluate the separation performance SI-SNR improvement (SI-SNRi) [7] as well as the standard source-to-distortion ratio (SDR) improvement (SDRi) [36] to measure the reconstructed signal quality. We also report the multiplier-accumulator operations (MACs) to measure the computational complexity of the models. The computational cost of the number of MACs per second was tested on 5 seconds of input audio. The network parameters (# Param) are reported in a million (M). The profiling was executed on a computer equipped with an NVIDIA GeForce RTX 3090 Ti graphics card.

A. Effects of the k in the Graph Encoder

On the WSJ0-2Mix dataset, under the window size $L = 20$, we tune the hyper-parameter k with the GE-ConvTasNet to explore the effects of the k latent representations' structural details in the graph encoder, and present the validation results in Table I. In Table I, we observe that the proposed GE-Conv-TasNet achieves the best separation performance in the case of $k = 20$.

B. Comparison With Baseline

In Table II, we report the performance of the Conv-TasNet systems with different window size L. We observe that by simply adding the graph encoder to the Conv-TasNet, we improve the separation performance for both $L = 20$ and $L = 40$ with similar number of parameters and MACs.

In addition, our GE-Conv-TasNet with $L = 40$ outperforms the Conv-TasNet with $L = 20$, and our GE-Conv-TasNet with $L = 20$ outperforms the Conv-TasNet with $L = 10$, which suggests that the proposed graph encoder is more effective than reducing the window size in terms of performance and computational complexity.

C. Benchmarking Performance

We use kernel size of 20, i.e. $L = 20$, in the convolutuinal encoders. The systems are denoted as 'Conv-TasNet-KS20' and 'DPRNN-KS20'. We report their comparison with other systems in the literature on WSJ0-2Mix dataset in Table III. We observe that GE-Conv-TasNet-KS20 and GE-DPRNN-KS20 outperform others, that validates the effectiveness of the proposed graph encoder. Table IV compares the GE-Conv-TasNet-KS20 and GE-DPRNN-KS20 with baselines and other systems on WHAM! and Libri2Mix benchmark. Observe from Table IV, our proposed GE-Conv-TasNet-KS20 achieves about 5 dB improvement on WHAM! than Con-TasNet-glN-KS20. The reason is that our proposed graph encoder collects the structural details of k embedding vectors around a embedding vector and inputs embedding vectors and these structural details together into the separation processing, leading to improving the separation performance. These results show the robustness of our proposed graph encoder for various datasets, which demonstrates that the GCN-based $\mathbf{A}_k$ provides the graph tools for considering the need of using more challenging and realistic datasets in future speech separation research.

TABLE III
SI-SNRi (dB) and SDRi (dB) for Different Methods on WSJ0-2Mix Dataset

Method	SI-SNRi	SDRi	# Param (M)
ADANet [38]	10.40	10.80	9.10
uPIT-BLSTM-ST [39]	–	10.00	92.70
DANet [7]	10.50	–	9.10
Chimera++ [40]	11.50	12.00	32.90
WA-MISI-5 [41]	12.60	13.10	32.90
LSTM-TasNet [42]	10.80	11.20	32.00
BLSTM-TasNet [42]	13.20	13.60	23.60
MP-GTF-KS20 [15]	15.93	16.20	8.71
Conv-TasNet-KS20 [7]	15.01	15.30	8.71
DPRNN-KS20 [33]	15.88	16.15	2.64
GE-Conv-TasNet-KS20 (Ours)	**16.28**	**16.55**	**8.78**
GE-DPRNN-KS20 (Ours)	**16.53**	**16.79**	**2.90**

TABLE IV
SI-SNRi (dB) for Different Methods on WHAM! and Libri2Mix Dataset

Dataset Method	WHAM!	Libri2Mix
Chimera++ [40], [43]	9.90	–
BLSTM-TasNet [43]	12.00	13.50
Learnable fbank [16]	12.90	–
Conv-TasNet-KS20 [7]	8.19	13.63
DPRNN-KS20 [33]	9.36	14.60
GE-Conv-TasNet-KS20 (Ours)	**13.02**	**14.25**
GE-DPRNN-KS20 (Ours)	**9.69**	**15.10**

VI. Conclusion

To overcome the high computation complexity arising from small window size in the time-domain speech separation, we introduce a graph representation $\mathbf{A}_k$ to the latent representation extraction. We confirm that the graph encoder is effective through a number of experiments. The experiments on WSJ0-2Mix, WHAM!, and Libri2Mix datasets show that the introduction of graph encoder consistently improves SI-SNRi and SDRi over the baselines. We will study on large scale speaker dataset as a future work.

References

[1] S. Getzmann, J. Jasny, and M. Falkenstein, "Switching of auditory attention in "cocktail-party" listening: ERP evidence of cueing effects in younger and older adults," *Brain Cogn.*, vol. 111, pp. 1–12, 2017.

[2] Z. Pan, R. Tao, C. Xu, and H. Li, "Selective listening by synchronizing speech with lips," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 1650–1664, 2022.

[3] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[4] X. Qian, M. Madhavi, Z. Pan, J. Wang, and H. Li, "Multi-target DoA estimation with an audio-visual fusion mechanism," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 4280–4284.

[5] R. Tao, Z. Pan, R. K. Das, X. Qian, M. Z. Shou, and H. Li, "Is someone speaking? Exploring long-term temporal features for audio-visual active speaker detection," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 3927–3935.

[6] Z. Pan, Z. Luo, J. Yang, and H. Li, "Multi-modal attention for speech emotion recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 364–368.

[7] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[8] Y. Luo and N. Mesgarani, "TasNet: Time-domain audio separation network for real-time, single-channel speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 696–700.

[9] Z. Pan, M. Ge, and H. Li, "A hybrid continuity loss to reduce over-suppression for time-domain target speaker extraction," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 1786–1790.

[10] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1003–1012.

[11] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[12] Z. Pan, M. Ge, and H. Li, "USEV: Universal speaker extraction with visual cue," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 3032–3045, 2022.

[13] Z. Pan, X. Qian, and H. Li, "Speaker extraction with co-speech gestures cue," *IEEE Signal Process. Lett.*, vol. 29, pp. 1467–1471, 2022.

[14] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 46–50.

[15] D. Ditter and T. Gerkmann, "A multi-phase gammatone filterbank for speech separation via TasNet," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 36–40.

[16] M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "Filterbank design for end-to-end speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6364–6368.

[17] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.

[18] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[19] A. Shirian, S. Tripathi, and T. Guha, "Dynamic emotion modeling with learnable graphs and graph inception network," *IEEE Trans. Multimedia*, vol. 24, pp. 780–790, 2022.

[20] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[21] J. Shi and J. M. Moura, "Graph signal processing: Modulation, convolution, and sampling," 2019, *arXiv:1912.06762*.

[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–14.

[23] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–15.

[24] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 7370–7377.

[25] F. Mao, X. Wu, H. Xue, and R. Zhang, "Hierarchical video frame sequence representation with deep convolutional graph network," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 262–270.

[26] P. Tzirakis, A. Kumar, and J. Donley, "Multi-channel speech enhancement using graph neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 3415–3419.

[27] A. Shirian and T. Guha, "Compact graph architecture for speech emotion recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 6284–6288.

[28] T. Wang, H. Guo, X. Yan, and Z. Yang, "Speech signal processing on graphs: The graph frequency analysis and an improved graph wiener filtering method," *Speech Commun.*, vol. 127, pp. 82–91, 2021.

[29] T. Wang, H. Guo, Q. Zhang, and Z. Yang, "A new multilayer graph model for speech signals with graph learning," *Digit. Signal Process.*, vol. 122, 2021, Art. no. 103360.

[30] G. Wichern et al., "WHAM!: Extending speech separation to noisy environments," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, G. Kubin and Z. Kacic, Eds., 2019, pp. 1368–1372.

[31] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "LibriMix: An open-source dataset for generalizable speech separation," 2020, *arXiv:2005.11262*.

[32] A. G. Marques, S. Segarra, and G. Mateos, "Signal processing on directed graphs: The role of edge directionality when processing and learning from network data," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 99–116, Nov. 2020.

[33] 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.

[34] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 12642–2646.

[35] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 4, pp. 787–796, Apr. 2018.

[36] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 10, pp. 1901–1913, Oct. 2017.

[37] Z.-Q. Wang, K. Tan, and D. Wang, "Deep learning based phase reconstruction for speaker separation: A trigonometric perspective," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 71–75.

[38] Z.-Q. Wang, J. L. Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 686–690.

[39] Z.-Q. Wang, J. L. Roux, D. Wang, and J. R. Hershey, "End-to-end speech separation with unfolded iterative phase reconstruction," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 2708–2712.

[40] Y. Luo and N. Mesgarani, "Real-time single-channel dereverberation and separation with time-domain audio separation network," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 342–346.

[41] Z. Yao, W. Pei, F. Chen, G. Lu, and D. Zhang, "Stepwise-refining speech separation network via fine-grained encoding in high-order latent domain," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 378–393, 2022.