TF-GridNet: Integrating Full- and Sub-Band Modeling for Speech Separation

Zhong-Qiu Wang, Samuele Cornell, Shukjae Choi, Younglo Lee, Byeong-Yeol Kim, and Shinji Watanabe, *Fellow, IEEE*

***Abstract*—We propose TF-GridNet for speech separation. The model is a novel deep neural network (DNN) integrating full- and sub-band modeling in the time-frequency (T-F) domain. It stacks several blocks, each consisting of an intra-frame full-band module, a sub-band temporal module, and a cross-frame self-attention module. It is trained to perform complex spectral mapping, where the real and imaginary (RI) components of input signals are stacked as features to predict target RI components. We first evaluate it on monaural anechoic speaker separation. Without using data augmentation and dynamic mixing, it obtains a state-of-the-art 23.5 dB improvement in scale-invariant signal-to-distortion ratio (SI-SDR) on WSJ0-2mix, a standard dataset for two-speaker separation. To show its robustness to noise and reverberation, we evaluate it on monaural reverberant speaker separation using the SMS-WSJ dataset and on noisy-reverberant speaker separation using WHAMR!, and obtain state-of-the-art performance on both datasets. We then extend TF-GridNet to multi-microphone conditions through multi-microphone complex spectral mapping, and integrate it into a two-DNN system with a beamformer in between (named as MISO-BF-MISO in earlier studies), where the beamformer proposed in this article is a novel multi-frame Wiener filter computed based on the outputs of the first DNN. State-of-the-art performance is obtained on the multi-channel tasks of SMS-WSJ and WHAMR!. Besides speaker separation, we apply the proposed algorithms to speech dereverberation and noisy-reverberant speech enhancement. State-of-the-art performance is obtained on a dereverberation dataset and on the dataset of the recent L3DAS22 multi-channel speech enhancement challenge.**

***Index Terms*—Acoustic beamforming, complex spectral mapping, full- and sub-band integration, speech separation.**

Manuscript received 22 November 2022; revised 9 July 2023; accepted 27 July 2023. Date of publication 11 August 2023; date of current version 25 August 2023. This work was part of the Delta Research Computing Project, which was supported in part by the National Science Foundation under Grant OCI 2005572 and the State of Illinois, Delta is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications, and in part by NVIDIA Corporation with the donation of the RTX 8000 GPUs. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Nobutaka Ito. *(Corresponding author: Zhong-Qiu Wang.)*

Zhong-Qiu Wang and Shinji Watanabe are with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: wang.zhongqiu41@gmail.com; shinjiw@cmu.edu).

Samuele Cornell is with the Department of Information Engineering, Universitá Politecnica delle Marche, 60121 Ancona, Italy (e-mail: cornellsamuele@gmail.com).

Shukjae Choi, Younglo Lee, and Byeong-Yeol Kim are with the Hyundai Motor Group and 42dot Inc., Seoul 06797, South Korea (e-mail: sjchoi.hmg@gmail.com; younglo.lee@42dot.ai; byeongyeol.kim@42dot.ai).

Digital Object Identifier 10.1109/TASLP.2023.3304482

I. Introduction

DEEP learning has dramatically advanced talker-independent speaker separation in the past decade [1], especially since deep clustering [2] and permutation invariant training (PIT) [3] successfully addressed the label permutation problem. Early studies train DNNs for magnitude estimation, with or without estimating phase [4], [5], [6], [7]. Subsequent studies carry out separation in the complex T-F domain via complex ratio masking [8] or in the time domain via TasNets [9], [10], [11]. Since 2019, Conv-TasNet and its variants [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], featuring advanced DNN architectures with learned encoder-decoder modules operating on very short windows of signals for end-to-end masking based separation, have gradually become the most popular and dominant approach for speaker separation in anechoic conditions. Their performance on the standard WSJ0-2mix benchmark [2] has reached an impressive SI-SDR improvement (SI-SDRi) of 22.1 dB [24].

In the meantime, T-F domain models, which usually use larger window and hop sizes, have been largely under-explored and under-represented in speaker separation in anechoic conditions. Recently, TFPSNet [25] reported a strong SI-SDRi of 21.1 dB on WSJ0-2mix, which is comparable to the top results achievable by modern time-domain models. It leverages a modern dual-path architecture, following DPRNN [15] and DPTNet [17], but applies the architecture on complex T-F spectrogram [26], [27] by using the transformer module proposed in DPTNet [17] to model spectro-temporal information. Although TFPSNet operates in the T-F domain [25], it closely follows the encoder-separator-decoder scheme [11] widely-used in TasNets and its performance, even with a modern DNN architecture, is still much lower than contemporary time-domain models [22], [23], [24].

In this context, for anechoic speaker separation our preliminary version [28] of this article made the following contributions to improve complex T-F domain approaches:

- We proposed to use complex spectral mapping for speaker separation in anechoic conditions. Complex spectral mapping [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], which predicts target RI components based on the RI components of input signals, has shown strong potential on noisy-reverberant speech separation when combined with modern DNN architectures and loss functions, exhibiting strong robustness to noise and reverberation in both single-

and multi-microphone conditions. Its potential on anechoic speaker separation, however, has not been studied, especially in an era when time-domain models, which perform masking in a learned filterbank domain, have become so popular and dominant on this task. This article is the first study to explore this direction for monaural, anechoic speaker separation.

- We proposed a novel DNN architecture named TF-GridNet for speech separation. It operates in the complex T-F domain to model speech spectrograms in a grid-like manner. Based on an improved TFPSNet [25], we add a cross-frame self-attention path for dual-path models to leverage global information across frames;
- Building upon the SI-SDR loss [11], [40], we proposed to add a novel loss term to encourage estimated sources to add up to the mixture. We also combine this loss term with loss functions other than SI-SDR.

Without using any data augmentation and dynamic mixing, on WSJ0-2mix [2] our best model obtains 23.5 dB SI-SDRi, clearly better than the previous best (at 22.1 dB) [24].

However, our preliminary study [28] does not show the potential of TF-GridNet for speech separation in noisy-reverberant conditions and it lacks an extension to multi-channel conditions. To address the first problem, we evaluate TF-GridNet on monaural reverberant speaker separation using the SMS-WSJ dataset [41] and monaural noisy-reverberant speaker separation using WHAMR! [42]. To address the second problem, we integrate TF-GridNet with a MISO-BF-MISO approach [34], [35], [36], which sandwiches a beamformer with two multi-channel-input single-channel-output (MISO) DNNs, with the beamformer computed based on the output of the first DNN and the second DNN performing post-filtering. In our recent work [43], we follow this MISO-BF-MISO approach and stack two TCN-DenseUNets with a novel multi-channel multi-frame Wiener filter (MFWF) in between. The TCN-DenseUNet [32], [33], [34], [35], [36], [37], [38], [39] is a strong, representative model adopted in many previous complex spectral mapping studies, and the MFWF is computed based on both DNN-estimated target magnitude and phase, and leverages both future and past frames for sub-band linear filtering. This solution won the recent L3DAS22 3D speech enhancement challenge [44], which attracted 17 submissions. In this article, a major difference from [43] is that we replace the TCN-DenseUNet with the newly-proposed TF-GridNet by modifying TF-GridNet for multi-microphone complex spectral mapping [34], [35], [36], and we observe large improvement over [43] and many other strong multi-channel systems. Both TF-GridNet and MISO-BF-MISO can be understood from the perspective of integrated full- and sub-band modeling, either inside TF-GridNet or outside through beamforming and post-filtering.

State-of-the-art performance is achieved on four major speech separation tasks, including reverberant speaker separation, noisy-reverberant speaker separation, speech dereverberation and noisy-reverberant speech enhancement, showing the effectiveness of the proposed algorithms at single- and multi-channel separation. In our experiments, for each task we strive to use public datasets with strong results published by previous studies. A sound demo is available online.[1] We have released the code of TF-GridNet in the ESPnet-SE++ toolkit [45].[2]

II. System Overview

A. Physical Model and Objective

For an N-sample, C-speaker mixture signal recorded by a P-microphones array in a noisy-reverberant setting, at sample n the physical model describing the relationship between the mixture $\mathbf{y}[n] \in \mathbb{R}^P$, reverberant non-target signals $\mathbf{v}[n] \in \mathbb{R}^P$, and dry source signal $(o(c))[n] \in \mathbb{R}$, direct-path signal $(\mathbf{s}(c))[n] \in \mathbb{R}^P$ and reverberation $(\mathbf{h}(c))[n] \in \mathbb{R}^P$ of speaker c can be formulated in the time domain as

$$\begin{aligned}\mathbf{y}[n] &= \sum_{c=1}^{C} \left(o(c) * \mathbf{r}(c)\right)[n] + \mathbf{v}[n] \\ &= \sum_{c=1}^{C} \left(\left(o(c) * \mathbf{r}^{d}(c)\right)[n] + \left(o(c) * \mathbf{r}^{e+l}(c)\right)[n]\right) + \mathbf{v}[n] \\ &= \sum_{c=1}^{C} \left(\left(\mathbf{s}(c)\right)[n] + \left(\mathbf{h}(c)\right)[n]\right) + \mathbf{v}[n], \end{aligned} \tag{1}$$

where $*$ is the linear convolution operator, and the P-channel room impulse response (RIR) of speaker c, $\mathbf{r}(c)$, can be decomposed into the RIR of the direct-path signal, $\mathbf{r}^{d}(c)$, and that of early reflections and late reverberation combined, $\mathbf{r}^{e+l}(c)$. In the short-time Fourier transform (STFT) domain, the physical model is formulated as

$$\mathbf{Y}(t,f) = \sum_{c=1}^{C} \left(\mathbf{S}(c,t,f) + \mathbf{H}(c,t,f)\right) + \mathbf{V}(t,f), \tag{2}$$

where t indexes T frames, f indexes F frequencies, and $\mathbf{Y}(t,f)$, $\mathbf{V}(t,f)$, and $\mathbf{S}(c,t,f)$ and $\mathbf{H}(c,t,f) \in \mathbb{C}^P$ respectively denote the STFT vectors of the mixture, non-target signals, and the direct-path signal and reverberation of speaker c. The corresponding spectrograms are denoted by $\mathbf{Y}$, $\mathbf{V}$, $\mathbf{S}(c)$, and $\mathbf{H}(c)$. This formulation covers all the tasks we consider:

- For monaural, anechoic speaker separation, $C > 1$, $P = 1$ and there is no $\mathbf{H}$ and $\mathbf{V}$;
- For reverberant speaker separation, $C > 1$ and $\mathbf{V}$ is a weak stationary noise (e.g., microphone sensor noise);
- For noisy-reverberant speaker separation, $C > 1$ and $\mathbf{V}$ consists of challenging non-stationary noises;
- For speech dereverberation, $C = 1$ and $\mathbf{V}$ is a weak stationary noise;
- For noisy-reverberant speech enhancement, $C = 1$ and $\mathbf{V}$ contains challenging non-stationary noises.

Given a single- or multi-channel mixture, we aim at reconstructing the direct-path signal of each speaker at a reference microphone q (i.e., $s_q(c)$). This requires us to not only remove noise and reverberation but also separate the speakers if there are more than one. For all the tasks, we assume that the maximum

[1] See https://zqwang7.github.io/demos/TF-GridNet-demo/index.html.
[2] See https://github.com/espnet/espnet/pull/5395

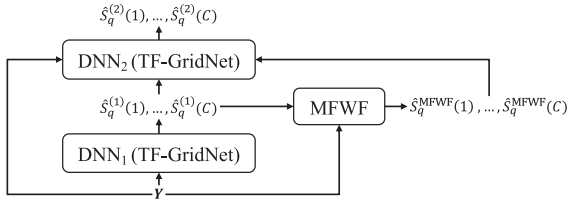

Fig. 1. System overview.

number of speakers in each mixture is known, and that the array geometry is fixed between training and testing.

B. Approach Outline

Our system (see Fig. 1) operates in the complex T-F domain. It follows a two-DNN approach named MISO-BF-MISO [35], [36], [37], where DNN$_1$ first produces an initial estimate for each target source, the initial estimate is then used to compute a sub-band linear filter (in this article a multi-frame Wiener filter) for each source, and DNN$_2$ takes in the mixture, the outputs of DNN$_1$, and the linear-filtering results for post-filtering. In our experiments, DNN$_1$ and DNN$_2$ are trained sequentially rather than jointly. After DNN$_1$ is trained, we use it to generate an initial estimate $\hat{S}_q^{(1)}(c)$ and compute a sub-band linear filtering result $\hat{S}_q^{\mathrm{MFWF}}(c)$ for each speaker c, and feed them and $\mathbf{Y}$ to DNN$_2$ to further predict target speech (denoted as $\hat{S}_q^{(2)}(c)$). The superscripts in $\hat{S}_q^{(1)}(c)$ and $\hat{S}_q^{(2)}(c)$ denote which of the two DNNs produces the estimate. Following [35], [37], [39], for speaker separation DNN$_1$ is trained with utterance-wise PIT [3] but DNN$_2$ is trained in an enhancement way (i.e, predicting all the speakers but not using PIT), since the label-permutation problem has been addressed by DNN$_1$. For monaural, anechoic speaker separation, we only train DNN$_1$, without using linear filtering and DNN$_2$.

III. TF-GridNet

Fig. 2 illustrates the proposed TF-GridNet for DNN$_2$. DNN$_1$ has the same architecture but uses only $\mathbf{Y}$ as input. Both DNNs are trained to perform complex spectral mapping [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], where the RI components of input signals are stacked as input features to predict the RI components of each speaker at the reference microphone q, i.e., $S_q(c)$. Our system is non-causal. We normalize the sample variance of each input mixture to 1.0 and use the same scaling factor to scale each target source before using them for training. This amounts to adjusting the volume of each input mixture to a similar level.

In Fig. 2, for each of the three real-valued input features (i.e., the stacked RI components of the mixture $\mathbf{Y}$ with shape $2P \times T \times F$, DNN$_1$'s outputs $\hat{S}_q^{(1)}(1),\ldots,\hat{S}_q^{(1)}(C)$ with shape $2C \times T \times F$, and MFWF's outputs $\hat{S}_q^{\mathrm{MFWF}}(1),\ldots,\hat{S}_q^{\mathrm{MFWF}}(C)$ with shape $2C \times T \times F$), we first use a two-dimensional (2D) convolution (Conv2D) with a 3×3 kernel followed by global layer normalization (gLN) [11] to compute a D-dimensional embedding for each T-F unit, and then summate the T-F embeddings generated for the three input features, obtaining a tensor with shape $D \times T \times F$. Next, we feed the tensor to B stacked TF-GridNet blocks, each consisting of an intra-frame full-band module, a sub-band temporal module, and a cross-frame self-attention module, to leverage spectral, spatial and temporal information to gradually make the T-F embeddings more discriminative for separation. After that, a 2D deconvolution (Deconv2D) with $2C$ output channels and a 3×3 kernel followed by linear units is used to obtain the predicted RI components for all the C speakers, and inverse STFT (iSTFT) is applied for signal re-synthesis. The rest of this section describes the three modules in each TF-GridNet block, and the loss functions. To avoid confusion, in Table I we list the hyper-parameters we will use to describe TF-GridNet.

TABLE I
List of Hyper-Parameters of TF-GridNet

Symbols	Description
D	Embedding dimension for each T-F unit
B	Number of TF-GridNet blocks
I	Kernel size for Unfold and Deconv1D
J	Stride size for Unfold and Deconv1D
H	Number of hidden units of BLSTMs in each direction
L	Number of heads in self-attention
E	Number of output channels in point-wise Conv2D to obtain key and query tensors in self-attention

A. Intra-Frame Full-Band Module

For the intra-frame module, we view the input tensor $R_b \in \mathbb{R}^{D\times T\times F}$ to the bth block as T separate sequences, each with length F, and use a sequence model to capture the full-band spectral and spatial information within each frame.

In detail, we first use the *torch.unfold* function [46] with kernel size I and stride J to stack nearby embeddings at each step along frequency, after zero-padding the frequency dimension to $F' = \lceil \frac{F-I}{J} \rceil \times J + I$, and then apply layer normalization (LN) along the first dimension, i.e.,

$$\dot{R}_b = \mathrm{LN}\left([\mathrm{Unfold}(R_b[:,t,:]), \text{ for } t = 1,\ldots,T]\right) \in \mathbb{R}^{(I\times D)\times T\times(\frac{F'-I}{J}+1)}. \quad (3)$$

We denote this order of operations as **Unfold-LN**. An alternative is to first perform LN on R_b and then zero-pad and stack nearby embeddings, i.e.,

$$\dot{R}_b = \left[\mathrm{Unfold}\left(\mathrm{LN}(R_b)[:,t,:]\right), \text{ for } t = 1,\ldots,T\right] \in \mathbb{R}^{(I\times D)\times T\times(\frac{F'-I}{J}+1)}. \quad (4)$$

We denote this order as **LN-Unfold**. We point out that LN-Unfold uses fewer parameters than Unfold-LN, and, since the *torch.unfold* function creates a view of the input tensor without allocating new memory, LN-Unfold consumes less memory when $I/J > 1$. Note that our preliminary paper [28] uses Unfold-LN, and this article proposes LN-Unfold, which leads to slightly better separation.

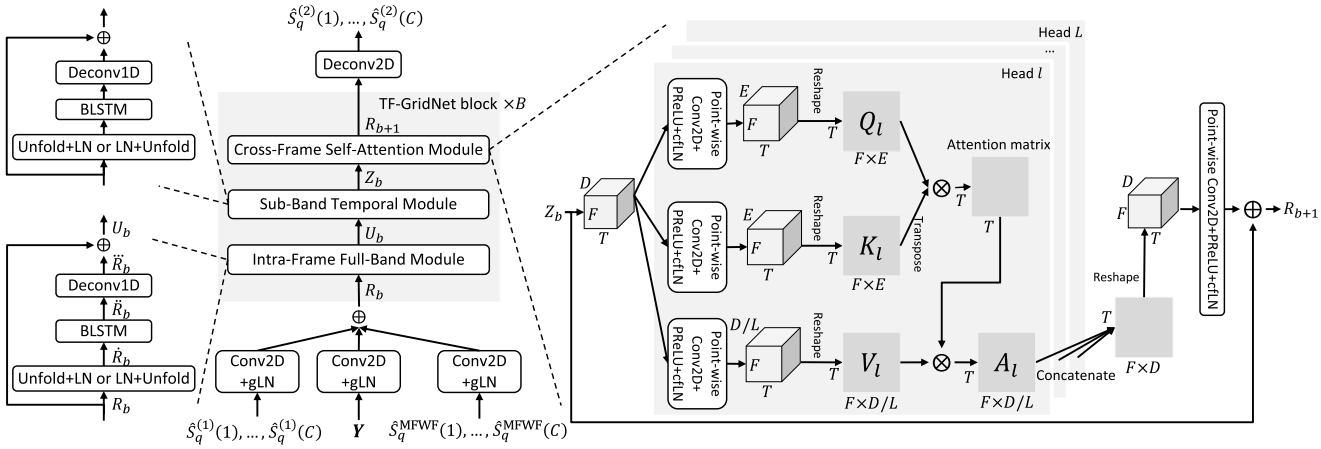

Fig. 2. Proposed TF-GridNet based DNN_2.

We then use a single bi-direcitonal long short-term memory (BLSTM) with H units in each direction to model inter-frequency information within each frame:

$$\ddot{R}_b = \Big[\text{BLSTM}\left(\text{LN}(\dot{R}_b)[:,t,:]\right), \quad \text{for } t = 1,\dots,T\Big] \in \mathbb{R}^{2H\times T\times(\frac{F'-I}{J}+1)}. \tag{5}$$

Note that J can be larger than one so that the sequence length and thus the amount of computation can be reduced.

Next, a one-dimensional deconvolution (Deconv1D) layer with kernel size I, stride J, input channel $2H$ and output channel D (and without subsequent normalization and non-linearity) is applied to the hidden embeddings of the BLSTM:

$$\dddot{R}_b = \Big[\text{Deconv1D}(\ddot{R}_b[:,t,:]), \quad \text{for } t = 1,\dots,T\Big] \in \mathbb{R}^{D\times T\times F'}. \tag{6}$$

After removing zero paddings, this tensor is added to the input tensor via a residual connection to produce the output tensor:

$$U_b = \dddot{R}_b[:,:,:F] + R_b \in \mathbb{R}^{D\times T\times F}. \tag{7}$$

B. Sub-Band Temporal Module

In the sub-band temporal module, the procedure is almost the same as that in the intra-frame full-band module. The only difference is that the input tensor $U_b \in \mathbb{R}^{D\times T\times F}$ is viewed as F separate sequences, each with length T, and a BLSTM is used to model the temporal information within each frequency. Note that the parameters of the BLSTM are shared across all the frequencies. The output tensor is denoted as $Z_b \in \mathbb{R}^{D\times T\times F}$.

C. Discussion on Full- and Sub-Band Modeling

In multi-channel conditions, performing sub-band modeling is a reasonable strategy to leverage spatial information afforded by multiple microphones. The idea is that inter-microphone spatial patterns such as the inter-channel phase differences (IPD) do not change along time for sources that do not move within each utterance, while they usually change with frequency due to the linear phase structure of phase differences and the effects of phase wrapping (see an example plot of IPD vs. frequency in anechoic conditions in Fig. 3 of [47]). This is partly the reason why many conventional beamforming [48], dereverberation [49] and spatial clustering [50] algorithms are performed separately within each frequency. In light of this physical phenomenon, we believe that it intuitively makes sense to perform such a DNN-based sub-band modeling, as the inter-channel phase patterns important for supervised learning are stable and salient within each frequency for each source. In addition, using a shared DNN block to separately model each sub-band is easier than using a DNN block to simultaneously model all the frequencies, as there are fewer variations to model. This echoes the idea of weight sharing, a core concept in convolutional neural networks [51].

Similarly, in multi-microphone conditions the intra-frame full-band module described in the previous subsection could not only model the full-band, spectral patterns such as the harmonic structure along frequency but also model the gradual changes of inter-microphone phase patterns along frequency (see the helix structure of IPD along frequency in Fig. 3(c) of [47]). We emphasize that the pattern of such gradual changes along frequency exists at every frame where the target source (assumed non-moving) is active. It is therefore reasonable to run the same BLSTM based full-band module at each frame to model such patterns.

Such sub-band modeling approach could better deal with reverberation. Since reverberation time (T60) and reverberation patterns vary with frequency [52], it is reasonable to use sub-band modules in TF-GridNet to separately model each frequency. In a broader perspective, weighted prediction error (WPE) [49], the most popular conventional algorithm for dereverberation, is also performed per-frequency by computing a linear, inverse filter at each frequency (preferably with different number of filter taps at different frequencies [53]) to estimate late reverberation. There are studies [54] using a non-linear LSTM to mimic the linear, inverse filtering of WPE, but the LSTM is trained to model all the frequencies simultaneously rather than separately. We believe that using sub-band DNN modules to mimic sub-band inverse filtering is likely better, because reverberation, at each frequency, can be approximated as a linear convolution of a sub-band filter and the anechoic signal,

according to the narrow-band approximation property [37], [48] in the STFT domain.

There are earlier studies using DNNs to perform full-band and sub-band modeling [55], [56], [57], [58]. Some differences include: (1) they only perform sub-band modelling without full-band modelling [55], [56]; and (2) they perform sub-band modeling followed by full-band modelling [57], [58] but without iterative information flow from sub- to full-band modules and from full- to sub-band, while we stack multiple TF-GridNet blocks to enable such an information flow so that full- and sub-band modelling can be integrated.

There are earlier studies [25], [59] using LSTMs to model spectrograms along time and frequency in monaural anechoic speaker separation. However, they do not reach very strong performance.

D. Cross-Frame Self-Attention Module

In the cross-frame self-attention module (shown in Fig. 2), we first compute frame embeddings at each frame using the T-F embeddings within that frame, and then use full-utterance self-attention on these frame embeddings to model long-range context information. The motivation is that the information flow between two T-F units needs to go through many steps in the intra-frame full-band and sub-band temporal BLSTMs, and the self-attention module enables each frame to directly attend to any frames of interest to allow for more direct information flow. We follow the self-attention mechanism proposed in [60], [61], which is designed for U-Net based monaural music source separation and speech denoising. In contrast, we use multi-head attention instead of single-head and we use the self-attention mechanism with the proposed sub-band and full-band modules rather than with U-Net for single- and multi-microphone speech separation.

The self-attention module has L heads. In each head l, we apply point-wise Conv2D, PReLU, LN along the channel and frequency dimensions (denoted as cfLN), and reshape layers to respectively obtain 2D query $Q_l \in \mathbb{R}^{T\times(F\times E)}$, key $K_l \in \mathbb{R}^{T\times(F\times E)}$ and value $V_l \in \mathbb{R}^{T\times(F\times D/L)}$ tensors. The point-wise Conv2D layers for computing the query and key tensors have E output channels, leading to $F \times E$-dimensional query and key vectors at each frame. Similarly, the point-wise Conv2D layer for computing the value tensor has D/L output channels, leading to an $F \times D/L$-dimensional value vector at each frame. All the three point-wise Conv2D layers has D input channels. Following [62], we compute the attention output $A_l \in \mathbb{R}^{T\times(F\times D/L)}$ by:

$$A_l = \text{softmax}\left(\frac{Q_l K_l^{\mathsf{T}}}{\sqrt{F\times E}}\right) V_l. \tag{8}$$

We then concatenate the attention outputs of all the L heads along the second dimension, reshape it back to $D \times T \times F$, apply a point-wise Conv2D with D input and D output channels followed by a PReLU and a cfLN to aggregate cross-head information. Next, we add it to the input tensor Z_b via a residual connection to obtain the output tensor R_{b+1}, which is fed to the next TF-GridNet block.

This self-attention mechanism only adds a negligible number of parameters by using point-wise Conv2D layers. It operates at the frame level and the memory cost on attention matrices is $\mathcal{O}(B \times L \times T^2)$. In comparison, TFPSNet [25] uses multi-head self-attention in each path-scanning module, and the memory cost on attention matrices is $\mathcal{O}(B \times L \times F \times T^2) + \mathcal{O}(B \times L \times T \times F^2)$, which is much higher.

E. Loss Functions

Since evaluation metrics usually change with datasets, we use different loss functions for different datasets, considering that different loss functions have their strengths and weaknesses [63]. This section describes two loss functions, SI-SDR and Wav+Mag, both defined based on the re-synthesized signals of predicted RI components. They have been proposed in earlier studies. Our novelty is a mixture-constraint loss term to be used with SI-SDR and Wav+Mag.

1) SI-SDR Loss With Mixture Constraint: For anechoic speaker separation, there is only DNN_1, without the linear-filtering module and DNN_2. The model in this case is trained with utterance-level PIT [3]. The loss function follows the SI-SDR loss [11], [40], but with two differences.

First, in the original SI-SDR metric paper [40], there are two definitions for SI-SDR. One scales *source* to equalize its gain with that of estimate, and the other instead scales *estimate*. The SI-SDR loss proposed in the seminal DANet [64] and Conv-TasNet [11] studies (and almost all the follow-up studies) uses the former, while our study uses the latter:

$$\mathcal{L}_{\text{SI-SDR-SE}} = -\sum_{c=1}^{C} 10\log_{10}\frac{\|s_q^{(c)}\|_2^2}{\|\hat{\alpha}_q^{(c)}\hat{s}_q^{(c)} - s_q^{(c)}\|_2^2}, \tag{9}$$

where $\|\cdot\|_2^2$ computes the L_2 norm, $\hat{s}_q^{(c)}$ is the re-synthesized signal based on the predicted RI components for speaker c, $\hat{\alpha}_q^{(c)} = \text{argmin}_\alpha \|\alpha\hat{s}_q^{(c)} - s_q^{(c)}\|_2^2 = (\hat{s}_q^{(c)})^{\mathsf{T}} s_q^{(c)} / (\hat{s}_q^{(c)})^{\mathsf{T}}\hat{s}_q^{(c)}$, and the "SE" in $\mathcal{L}_{\text{SI-SDR-SE}}$ means "scaling estimate". We observe that this loss leads to similar performance and faster convergence, compared with the former.

Second, we add a loss term between the summation of target sources and that of scaled estimated sources:

$$\mathcal{L}_{\text{SI-SDR-SE+MC}} = \mathcal{L}_{\text{SI-SDR-SE}} + \frac{1}{N}\left\|\sum_{c=1}^{C}\hat{\alpha}_q^{(c)}\hat{s}_q^{(c)} - \sum_{c=1}^{C} s_q^{(c)}\right\|_1, \tag{10}$$

where $\|\cdot\|_1$ computes the L_1 norm and N denotes the number of samples. Since $y_q = \sum_{c=1}^{C} s_q^{(c)}$ in our considered task of monaural, anechoic speaker separation, we name the loss term as mixture-constraint (MC) loss. It is motivated by a trigonometric perspective [7] in source separation, which suggested that constraining the separated sources to sum up to the mixture yields better phase estimation. We point out that $\sum_{c=1}^{C}\hat{\alpha}_q^{(c)}\hat{s}_q^{(c)}$ would not equal y_q at run time. This distinguishes our loss from mixture consistency [65], which enforces the separated sources to sum up to the mixture. Our loss is also different from another mixture consistency loss proposed in [66], where the DNN is

trained for real-valued phase-sensitive masking without phase estimation and the task is target speaker extraction based meeting transcription.

In (10), we do not include a weighting term between the two losses for two reasons. First, this can avoid a weighting term to tune. Second, nowadays it is common for speaker separation systems to obtain more than 10 dB SI-SDRi, and when the sample variance of the input mixture has been normalized to 1.0 (which is the case in our study), the second term in our experiments has a scale less than 0.01 when the models converge. This way, the first term dominates the combined loss. This is desirable as the first term is directly related to the final separation performance.

2) Wav+Mag Loss: Following [63], we define the loss on the re-synthesized signal and its magnitude:

$$\mathcal{L}_{\text{Wav+Mag}} = \sum_{c=1}^{C} \left(\frac{1}{N} \|\hat{s}_q(c) - s_q(c)\|_1 + \frac{1}{T \times F} \left\| \left| \text{STFT}(\hat{s}_q(c)) \right| - \left| \text{STFT}(s_q(c)) \right| \right\|_1 \right), \tag{11}$$

where $|\cdot|$ computes magnitude and $\text{STFT}(\cdot)$ extracts a complex spectrogram. It has been demonstrated in [63] that the magnitude loss can improve metrics such as perceptual evaluation of speech quality (PESQ), short-time objective intelligibility [67] (STOI), and word error rates (WER) which favor signals with a good magnitude, at a degradation on time-domain metrics such as SI-SDR. When $C > 1$, we can also add a mixture-constraint loss, similarly to (10):

$$\begin{aligned}\mathcal{L}_{\text{Wav+Mag+MC}} = \sum_{c=1}^{C} \Bigg(& \frac{1}{N} \|\hat{s}_q(c) - s_q(c)\|_1 \\ & + \frac{1}{T \times F} \left\| \left| \text{STFT}(\hat{s}_q(c)) \right| - \left| \text{STFT}(s_q(c)) \right| \right\|_1 \Bigg) \\ & + \frac{1}{N} \left\| \sum_{c=1}^{C} \hat{s}_q(c) - \sum_{c=1}^{C} s_q(c) \right\|_1 \\ & + \frac{1}{T \times F} \left\| \left| \text{STFT} \left(\sum_{c=1}^{C} \hat{s}_q(c) \right) \right| \right. \\ & \left. - \left| \text{STFT} \left(\sum_{c=1}^{C} s_q(c) \right) \right| \right\|_1 . \end{aligned} \tag{12}$$

In (11) and (12), we do not use a weighting term, as the time-domain loss and the frequency-domain loss are on a similar scale due to the Parseval's theorem.

IV. Beamforming and Sub-Band Modelling

This section proposes a novel DNN-supported beamformer and connects it with integrated sub- and full-band modeling.

A. DNN-Supported Multi-Frame Wiener Filter

Assuming that target speakers are non-moving within each utterance and based on the estimated target speech $\hat{S}_q^{(1)}(c)$ by DNN_1, we compute a time-invariant MFWF per frequency by solving the minimization problem below:

$$\underset{\mathbf{w}_q(c,f)}{\text{argmin}} \sum_{t=1}^{T} \left| \hat{S}_q^{(1)}(c,t,f) - \mathbf{w}_q(c,f)^{\mathsf{H}} \widetilde{\mathbf{Y}}(t,f) \right|^2, \tag{13}$$

where $\widetilde{\mathbf{Y}}(t,f) = [\mathbf{Y}(t-\Delta_l,f)^{\mathsf{T}}, \ldots, \mathbf{Y}(t,f)^{\mathsf{T}}, \ldots, \mathbf{Y}(t+\Delta_r,f)^{\mathsf{T}}]^{\mathsf{T}}$ stacks the mixtures at nearby T-F units, $\mathbf{w}_q(c,f) \in \mathbb{C}^{(\Delta_l+1+\Delta_r)\times P}$ is a time-invariant linear filter, and $(\cdot)^{\mathsf{H}}$ computes complex Hermitian. Δ_l (≥ 0) and Δ_r (≥ 0) control the context of frames for filtering, resulting in a single-frame Wiener filter when Δ_l and Δ_r are both zeros and an MFWF otherwise. A closed-form solution is available:

$$\begin{aligned}&\hat{\mathbf{w}}_q(c,f) \\ &= \left(\sum_{t=1}^{T} \widetilde{\mathbf{Y}}(t,f) \widetilde{\mathbf{Y}}(t,f)^{\mathsf{H}} \right)^{-1} \sum_{t=1}^{T} \widetilde{\mathbf{Y}}(t,f) \left(\hat{S}_q^{(1)}(c,t,f) \right)^*, \end{aligned} \tag{14}$$

where $(\cdot)^*$ computes complex conjugate. The filtering result $\hat{S}_q^{\text{MFWF}}(c)$ is computed as

$$\hat{S}_q^{\text{MFWF}}(c,t,f) = \hat{\mathbf{w}}_q(c,f)^{\mathsf{H}} \widetilde{\mathbf{Y}}(t,f). \tag{15}$$

We name MFWF as MCMFWF when $P > 1$ and as single-channel MFWF (SCMFWF) when $P = 1$.

The idea of MCMFWF was proposed in [68]. Differently, we use multi-microphone complex spectral mapping to obtain $\hat{S}_q^{(1)}(c)$, which consists of DNN-estimated magnitude and phase, while the system in [68], even in multi-microphone cases, performs monaural, real-valued magnitude masking to obtain $\hat{S}_q^{(1)}(c)$, which consists of DNN-estimated magnitude and the mixture phase. It should be noted that in our recent studies [36], [69], we proposed to project the mixture to DNN-estimated target speech using (13), but the beamformer is single-frame (i.e., $\Delta_l = 0$ and $\Delta_r = 0$). We will show in our experiments that single-frame filtering leads to worse performance than multi-frame filtering, likely due to its insufficient degrees of freedom for suppressing non-target signals.

In monaural conditions, (14) becomes an SCMFWF, which can reduce reverberation by exploiting the correlations among nearby frames due to reverberation. It is similar to the inverse convolutive prediction filter proposed in [37]. The key different is that, in [37], only past frames are filtered (i.e., $\Delta_l > 0$ and $\Delta_r = 0$). However, future frames are also correlated with the current frame and they can also be linear-filtered to reduce the reverberation at the current frame.

In the literature, convolutional beamformer [53] and WPE [49] are the most popular multi-frame linear filters. In their DNN-supported versions, DNN-estimated target magnitude is used in a maximum-likelihood objective for filter computation [70]. We will show in our experiments that the output of the proposed MFWF improves the performance of DNN_2 by a larger factor.

B. Discussion on Beamforming and Sub-Band Modelling

When beamforming results are used as extra features for DNN training (e.g., in the way shown in Fig. 1), large improvement

has been observed in earlier studies [35], [36] (see for example the last two rows of Table XI). One interesting observation is that the DNNs in these studies usually perform full-band modelling, where one typical approach is to use an encoder to encode each frame into an embedding, perform sequence modelling to refine the frame embeddings, and use a decoder to reconstruct target speech from the refined embeddings. The encoder, for example, can be just a linear fully-connected layer followed by a non-linear activation [11] or contain a stack of non-linear layers in the form of a UNet-style encoder [35]. Our insight is that the large improvement is likely because the beamformers are computed based on signals only within each sub-band and the beamforming results could hence be complementary to full-band modeling, which simultaneously models all the frequencies but may not be good at sufficiently modeling each frequency since different frequencies exhibit diverse spectral, temporal and spatial patterns (see also our discussions in Section III-C).

Each sub-band temporal module in TF-GridNet models each frequency using a BLSTM shared across all the frequencies to mimic sub-band filtering. This could be a better way of *neural beamforming* than earlier approaches where DNNs are mainly used for full-band modeling. In our best-performing system, we still compute an MCMFWF result based on the output of a first TF-GridNet and use a second TF-GridNet for post-filtering (i.e., Fig. 1). This can be viewed as another way of full- and sub-band integrated modeling, and is found to improve the performance of using just one single TF-GridNet, but the improvement brought by the beamformer followed by post-filtering is much less impressive than the one achieved when the two DNNs are trained to perform full-band modelling. See also our discussion later in Section VI-C.

We point out that the sub-band (*a.k.a* narrow-band) property for per-frequency modeling is afforded by STFT. This property bears an important advantage of STFT-domain approaches: we can exploit intra- and cross-frequency information to achieve potentially better separation. In comparison, the learned bases by time-domain models are usually not narrow-band [11], [71], and many current time-domain models do not have a concept of sub-band or narrow-band frequency to exploit.

V. Experimental Setup

We evaluate the proposed algorithms on five tasks, including speaker separation in anechoic, reverberant and noisy-reverberant conditions, speech dereverberation, and noisy-reverberant speech enhancement. This section describes the setup for each task, baselines, and miscellaneous configurations. Our experiments cover major speech separation tasks and we use public datasets with existing published results to highlight that the improvements obtained in our study are relative to very strong baselines.

A. Setup for Monaural, Anechoic Speaker Separation

We use **WSJ0-2mix** [2], the most popular dataset to benchmark monaural talker-independent speaker separation algorithms in anechoic conditions. It has 20,000 (∼30.4 h), 5,000 (∼7.7 h) and 3,000 (∼4.8 h) two-speaker mixtures respectively in its training, validation and test sets. The clean source signals are sampled from the WSJ0 corpus. The speakers in the training and validation sets are different from the speakers for testing. The two utterances in each of the mixtures available in WSJ0-2mix are fully-overlapped, and their relative energy level is uniformly sampled from the range $[-5, 5]$ dB when WSJ0-2mix is created. The sampling rate is 8 kHz.

B. Setup for Reverberant Speaker Separation

We use **SMS-WSJ** [41], a popular corpus for comparing two-speaker separation algorithms in reverberant conditions. The clean speech is sampled from the WSJ0 and WSJ1 datasets. The corpus contains 33,561 (∼87.4 h), 982 (∼2.5 h) and 1,332 (∼3.4 h) two-speaker mixtures for training, validation and testing, respectively. The simulated microphone array has six microphones arranged uniformly on a circle with a diameter of 20 cm. For each mixture, the speaker-to-array distance is drawn from the range $[1.0, 2.0]$ m, and T60 from $[0.2, 0.5]$ s. A weak white noise is added to simulate microphone sensor noises, and the energy level between the sum of the reverberant speech signals and the noise is sampled from the range $[20, 30]$ dB. The sampling rate is 8 kHz.

For ASR evaluation, the default Kaldi-based ASR backend provided with SMS-WSJ [41] is used. It is trained using single-speaker noisy-reverberant speech as inputs and the state alignments of its corresponding direct-path signal as labels. A standard tri-gram language model is used for decoding.

We perform joint denoising, dereverberation and separation. We consider one-, two- and six-channel tasks, and use the direct-path signals as the training target. For two-channel processing, we take the signals at microphone 1 and 4 as input, and for monaural separation, we use the signal at microphone 1. The first microphone is always used as the reference.

C. Setup for Noisy-Reverberant Speaker Separation

We use **WHAMR!** [42] to validate our algorithms for noisy-reverberant speaker separation. It re-uses the two-speaker mixtures in WSJ0-2mix [2] but reverberates each clean source and adds non-stationary noises. In each mixture, the T60 is sampled from the range $[0.2, 1.0]$ s, signal-to-noise ratio (SNR) between the louder speaker and noise from $[-6, 3]$ dB, relative energy level between the two speakers from $[-5, 5]$ dB, and speaker-to-array distance from $[0.66, 2.0]$ m. There are 20,000 (∼30.4 h), 5,000 (∼7.7 h) and 3,000 (∼4.8 h) binaural mixtures respectively for training, validation and testing. We use its *min* and 8 kHz version.

We aim at joint dereverberation, denoising and speaker separation. The direct-path signal of each speaker at the first microphone is used as the target for training and as the reference for metric computation.

D. Setup for Speech Dereverberation

We use a simulated reverberant dataset with weak air-conditioning noises, since there lacks a well-designed popular dataset for speech dereverberation.[3] Although simulated by

[3]We considered the REVERB corpus [72], but its training set is simulated based on 24 eight-channel RIRs, which are too few for training DNN models.

ourselves, this dataset has been used in our recent studies [36], [37], which reported very strong results. The clean source signals for simulation are from the WSJCAM0 corpus, which includes 7,861, 742 and 1,088 utterances respectively in its training, validation and test sets. Based on them, we simulate 39,293 (∼77.7 h), 2,968 (∼5.6 h), and 3,262 (∼6.4 h) noisy-reverberant mixtures respectively as our training, validation, and test sets. The data spatialization process follows [34], where, for each utterance, we randomly sample a room with random room characteristics and speaker and microphone locations, using the Pyroomacoustics RIR generator [73]. The simulated microphone array has eight microphones arranged on a circle with a diameter of 20 cm. The speaker-to-array distance is drawn from the range $[0.75, 2.5]$ m and T60 from $[0.2, 1.3]$ s. For each utterance, an eight-channel diffuse air-conditioning noise is sampled from the REVERB dataset [72] and added to the reverberant speech, and the SNR between the direct-path signal and the noise is sampled from the range $[5, 25]$ dB. The sampling rate is 16 kHz. We denote this dataset as **WSJ0CAM-DEREVERB**.

We aim at removing any early reflections and late reverberation. The direct-path signal of the target speaker at the first microphone is used as the reference for metric computation.

E. Setup for Noisy-Reverberant Speech Enhancement

The **L3DAS22** 3D speech enhancement task [44] challenges participants to reconstruct the dry speech source signal from its far-field mixture simulated by using two four-channel Ambisonic-format signals in a noisy-reverberant office environment. The dry source signals are drawn from LibriSpeech and noise signals from FSD50k [74]. The SNR is sampled from the range $[6, 16]$ dB. Real RIRs are used for simulation. Such RIRs were recorded in an office room by using two first-order A-format Ambisonic arrays, each with four microphones. The microphone placement is fixed, with one Ambisonic microphone array placed at the room center and the other being 20 cm away. The room configuration is the same between training and testing, and the source positions are sampled uniformly inside the room with no overlap of positions between training and testing. Artificial mixtures are generated by convolving dry speech and dry noise signals with the measured RIRs and the convolved signals are then added together. There are 37,398 (∼81.3 h), 2,362 (∼3.9 h) and 2,189 (∼3.5 h) mixtures respectively in the training, validation and test sets. The generated A-format Ambisonic mixtures are converted to B-format Ambisonic via a transformation consisting of a pre-filter, a mixing matrix and a post-filter. The task is to predict the dry speech based on the B-format Ambisonic mixture. The sampling rate is 16 kHz.

The submitted systems were ranked by using a combination of STOI and WER:

$$\text{Task1Metric} = (\text{STOI} + (1 - \text{WER}))/2. \tag{16}$$

Since STOI and WER scores are both in the range of [0, 1], the composite metric is also in [0, 1]. The WER is computed from the transcription of enhanced speech with that of the dry speech, both decoded by a pre-trained wav2vec2 ASR model.

Differently from the other setups, the goal in this task is to predict the dry speech from far-field multi-channel mixtures. This requires the submitted systems to not only remove reverberation and noises, but also to time-align the estimated speech with the dry speech (as STOI degrades with misalignment), which requires the systems to perform implicit or explicit localization of the target source so that a time-aligned estimate can be obtained. This is achievable since the Ambisonic arrays form a fixed three-dimensional geometry.

F. Baselines

We can compare our approaches with others by using system-level performance. For MFWF, we provide the results of other linear filters, including (1) in multi-channel cases, convolutional beamformer [53]; and (2) in monaural cases, WPE [49], [70]. We replace the MFWF module between DNN_1 and DNN_2 in Fig. 1 with a DNN-supported convolutional beamformer or WPE filter to compare their effectiveness at improving DNN_2.

1) System-Level Baselines: Since the datasets in all the considered tasks have existing results reported in earlier studies, we can compare our results with the strongest ones achieved by competing approaches. Notably, we will compare with our previous studies [35], [37], [39], [43], which also follow the MISO-BF-MISO approach shown in Fig. 1 but uses TCN-DenseUNet and other sub-band linear filters.

2) Baseline for MCMFWF: In multi-channel cases, we consider convolutional beamformer [53], a very popular multi-channel multi-frame filter in speech separation, as the baseline. We compute it by solving the problem [53] below:

$$\begin{aligned} &\underset{\mathbf{w}_q(c,f)}{\operatorname{argmin}} \sum_{t=1}^{T} \frac{|\mathbf{w}_q(c,f)^{\mathsf{H}}\, \bar{\mathbf{Y}}(t,f)|^2}{\hat{\lambda}_q(c,t,f)} \\ &\text{subject to} \quad \mathbf{w}_{q;0}(c,f)^{\mathsf{H}} \hat{\mathbf{d}}_q(c,f) = 1, \end{aligned} \tag{17}$$

where $\bar{\mathbf{Y}}(t,f) = [\mathbf{Y}(t-\Delta_d-\Delta_l+1,f)^{\mathsf{T}}, \ldots, \mathbf{Y}(t-\Delta_d,f)^{\mathsf{T}}, \mathbf{Y}(t,f)^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{C}^{(\Delta_l+1)\times P}$ with Δ_d denoting a prediction delay and Δ_l the number of filter taps for past frames beyond the prediction delay, $\mathbf{w}_q(c,f) = [\mathbf{w}_{q;-\Delta_d-\Delta_l+1}(c,f)^{\mathsf{T}}, \ldots, \mathbf{w}_{q;-\Delta_d}(c,f)^{\mathsf{T}}, \mathbf{w}_{q;0}(c,f)^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{C}^{(\Delta_l+1)\times P}$ with $\mathbf{w}_{q;i}(c,f) \in \mathbb{C}^P$ denoting the filter applied to frame $t+i$ in order to produce the result at the current frame t, and $\hat{\mathbf{d}}_q(c,f)$ is the estimated relative transfer function for microphone q. Following [75] and based on the DNN-estimated target speech $\hat{S}_q^{(1)}(c)$, $\hat{\lambda}_q(c)$, the estimated power spectral density of target speech, can be computed as:

$$\hat{\lambda}_q(c,t,f) = \max\left(\varepsilon\, \max(|\hat{S}_q^{(1)}(c)|^2), |\hat{S}_q^{(1)}(c,t,f)|^2\right), \tag{18}$$

where $\max(\cdot)$ extracts the maximum value of a spectrogram, $\max(\cdot,\cdot)$ returns the larger of two values, and ε is a floor value to avoid putting too much weight on T-F units with low energy. Through T-F masking and also based on the DNN-estimated target speech $\hat{S}_q^{(1)}(c)$, $\hat{\mathbf{d}}_q(c,f)$ is computed as the principal eigenvector of an estimated speech covariance matrix [48], [76],

[77] for non-moving point sources, i.e.,

$$\hat{\mathbf{\Phi}}(c,f) = \sum_{t=1}^{T} \hat{m}(c,t,f)\mathbf{Y}(t,f)\mathbf{Y}(t,f)^{\mathsf{H}}, \tag{19}$$

$$\hat{m}(c,t,f) = \frac{|\hat{S}_q^{(1)}(c,t,f)|}{|\hat{S}_q^{(1)}(c,t,f)| + |Y_q(t,f) - \hat{S}_q^{(1)}(c,t,f)|}, \tag{20}$$

$$\hat{\mathbf{d}}(c,f) = \mathcal{P}\left(\hat{\mathbf{\Phi}}(c,f)\right), \tag{21}$$

$$\hat{\mathbf{d}}_q(c,f) = \hat{\mathbf{d}}(c,f)/\hat{d}(c,f;q), \tag{22}$$

where $\mathcal{P}(\cdot)$ extracts the principal eigenvector, and $\hat{d}(c,f;q)$ denotes the qth element in $\hat{\mathbf{d}}(c,f)$. The results of convolutional beamformer is comptued as

$$\hat{S}_q^{\text{ConvBF}}(c,t,f) = \hat{\mathbf{w}}_q(c,f)^{\mathsf{H}}\, \bar{\mathbf{Y}}(t,f), \tag{23}$$

where "ConvBF" denotes convolutional beamformer.

Notice that our DNN-supported MCMFWF in (13) is simpler to compute than the convolutional beamformer.

3) Baseline for SCMFWF: In the single-microphone case, convolutional beamformer turns into the WPE filter [49]. Following the DNN-WPE algorithm [70], we compute it by using the magnitude of $\hat{S}_q^{(1)}(c)$ estimated by DNN_1. The filter is computed by solving the following problem:

$$\underset{\mathbf{w}_q(c,f)}{\text{argmin}} \sum_{t=1}^{T} \frac{|Y_q(t,f) - \mathbf{w}_q(c,f)^{\mathsf{H}}\, \breve{\mathbf{Y}}(t-\Delta_d,f)|^2}{\hat{\lambda}_q(c,t,f)}, \tag{24}$$

where $\breve{\mathbf{Y}}_q(t,f) = [Y_q(t-\Delta_l+1,f)^{\mathsf{T}}, \ldots, Y_q(t,f)^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{C}^{\Delta_l}$, $\mathbf{w}_q(c,f) \in \mathbb{C}^{\Delta_l}$, Δ_d is a prediction delay, and $\hat{\lambda}_q(c)$ is computed using (18). The WPE result is obtained as

$$\hat{S}_q^{\text{WPE}}(c,t,f) = Y_q(t,f) - \hat{\mathbf{w}}_q(c,f)^{\mathsf{H}}\breve{\mathbf{Y}}(t-\Delta_d,f). \tag{25}$$

G. Miscellaneous Setup

In default, for STFT, the window length is 32 ms and hop length 8 ms, and the square-root Hann window is used as the analysis window. In this case, for 16 kHz sampling rate, a 512-point discrete Fourier transform (DFT) is applied to extract 257-dimensional complex STFT spectra at each frame, and for 8 kHz, a 256-point DFT is used to extract 129-dimensional complex STFT spectra. E (see its definition in Table I) is set to 4 for 8 kHz and to 2 for 16 kHz. This way, the dimension of frame-level embeddings (i.e., $F \times E$) used for self-attention is reasonable.

For MFWF, we set Δ_l and Δ_r, which controls the filter taps, to 4 and 3 for eight-channel separation, to 5 and 4 for six-channel, to 15 and 14 for two-channel, and to 20 and 19 for single-channel. For convolutional beamformer, we set the prediction delay Δ_d to 3 following [53], and tune Δ_l to 7 for eight-channel processing, to 9 for six-channel, and to 29 for two-channel. For WPE, Δ_d is also 3 and Δ_l is tuned to 40. We emphasize that a positive prediction delay Δ_d is found important for convolutional beamformer and WPE to avoid target cancellation [49], [53], and both filters are designed by the original authors to not filter future frames, because future frames contain the reverberation of the target speech at the current frame and including them for linear filtering would lead to target cancellation. ε in (18) is tuned to 10^{-5}.

In each epoch, we sample a 4-second segment from each mixture for training. We normalize the sample variance of each mixture segment to 1.0 and use the same scaling factor to scale the target sources, before using them for training. Adam is used as the optimizer. The L_2 norm for gradient clipping is set to 1.0. The learning rate starts from 0.001 and is reduced by half if the validation loss does not improve in 3 epochs.

We do not use any dynamic mixing or data augmentation [19].

H. Evaluation Metrics

The evaluation metrics vary with tasks. We consider SI-SDR or SI-SDRi [40], SDR or SDR improvement (SDRi) [78], PESQ, STOI or extended STOI (eSTOI) [67][4], and WER. For PESQ, we use the *python-pesq* toolkit[5] to report narrow-band MOS-LQO scores. SI-SDR and SDR measure the quality of predictions at the sample level, PESQ and STOI are objective metrics of speech quality and intelligibility respectively, and WER is a widely-used metric for measuring speech recognition performance.

The number of model parameters is reported in millions (M). We use the *flops-counter.pytorch* toolkit[6] to count the number of multiply–accumulate (MAC) operations needed to process a 4-second mixture, and report it in giga-operations per second (GMAC/s). Following [79], we implement Deconv1D as a linear layer followed by overlap-add. This can reduce the number of MAC operations when the stride J is larger than 1, and speed up training and inference when the kernel size I equals J (> 1).

VI. Evaluation Results

We first show the effectiveness of TF-GridNet at separation on various tasks and datasets, and then present a study on the computational requirements of different TF-GridNet configurations and their performance on WSJ0-2mix.

A. Results on WSJ0-2mix

We evaluate TF-GridNet on monaural, anechoic speaker separation. SI-SDRi [40] and SDRi [78] are used as the evaluation metrics, following previous studies. The mixture SI-SDR is 0 dB and the mixture SDR 0.2 dB. We always use $B = 6$ blocks for WSJ0-2mix.

1) Comparison With DPRNN: Table II compares the performance of TF-GridNet with DPRNN [15]. We configure TF-GridNet to use almost the same number of parameters and almost the same amount of computation as DPRNN. This is implemented by using BLSTMs in each model and unifying the embedding dimension (and the bottleneck dimension in the cases of DPRNN) to 64 and the hidden dimension of the BLSTMs to 128. For DPRNN, we set the window size to 2 samples, hop

[4]https://github.com/mpariente/pystoi, v0.3.3

[5]https://github.com/ludlows/python-pesq, v0.0.2

[6]https://github.com/sovrasov/flops-counter.pytorch. Note that, in default, *flops-counter.pytorch* only tries to count the MAC operations of a list of predefined modules that are already available in Pytorch. We confirm that we also count the MAC operations of our customized modules.

TABLE II
MASKING AND MAPPING COMPARISON BASED ON WSJ0-2MIX

Row	Systems	Unfold+LN or LN+Unfold	Use attention?	Masking or Mapping?	Window/hop sizes (ms)	D	I	J	H	#params (M)	GMAC/s	Loss	SI-SDRi (dB)
1	DPRNN [15]	-	-	Embedding masking	0.25/0.125	-	-	-	-	2.6	42.2	(9)	18.8
2	TF-GridNet	Unfold+LN	✗	Embedding masking	32/8	64	1	1	128	2.8	47.6	(9)	20.7
3	TF-GridNet	Unfold+LN	✗	Complex ratio masking	32/8	64	1	1	128	2.6	42.4	(9)	20.8
4	TF-GridNet	Unfold+LN	✗	Complex spectral mapping	32/8	64	1	1	128	2.6	42.4	(9)	**21.2**

TABLE III
ABLATION RESULTS ON WSJ0-2MIX

Row	Systems	Unfold+LN or LN+Unfold	Use attention?	L	D	I	J	H	#params (M)	Loss	SI-SDRi (dB)
1	TF-GridNet	Unfold+LN	✗	-	64	1	1	128	2.6	(9)	21.2
2	TF-GridNet	Unfold+LN	✗	-	16	4	1	128	2.6	(9)	20.5
3	TF-GridNet	Unfold+LN	✗	-	128	1	1	128	3.6	(9)	21.6
4	TF-GridNet	Unfold+LN	✗	-	16	8	1	128	3.6	(9)	21.6
5	TF-GridNet	Unfold+LN	✗	-	16	8	1	128	3.6	(10)	21.8
6	TF-GridNet	Unfold+LN	✗	-	16	8	1	192	6.5	(10)	21.9
7	TF-GridNet	Unfold+LN	✗	-	24	8	1	192	8.0	(10)	22.5
8	TF-GridNet	Unfold+LN	✓	1	24	8	1	192	8.0	(10)	22.6
9	TF-GridNet	Unfold+LN	✓	4	24	8	1	192	8.1	(10)	22.9
10	TF-GridNet	Unfold+LN	✓	4	48	4	1	192	8.2	(10)	23.0
11	TF-GridNet	LN+Unfold	✓	4	48	4	1	192	8.2	(10)	23.2
12	TF-GridNet	LN+Unfold	✓	4	64	4	1	256	14.5	(10)	**23.5**

TABLE IV
PERFORMANCE COMPARISON WITH OTHER SYSTEMS ON WSJ0-2MIX

Systems	Domain	Year	#params (M)	SI-SDRi (dB)	SDRi (dB)
DPCL++ [4]	T-F	2016	13.6	10.8	-
uPIT-BLSTM-ST [3]	T-F	2017	92.7	-	10.0
ADANet [64]	T-F	2018	9.1	10.4	10.8
WA-MISI-5 [6]	T-F	2018	32.9	12.6	13.1
Sign Prediction Net [7]	T-F	2019	56.6	15.3	15.6
Conv-TasNet [11]	Time	2019	5.1	15.3	15.6
Deep CASA [8]	T-F	2019	12.8	17.7	18.0
Conv-TasNet-MBT [12]	Time	2020	8.8	15.6	-
FurcaNeXt [13]	Time	2020	51.4	-	18.4
SUDO RM -RF [14]	Time	2020	2.6	18.9	-
DPRNN [15]	Time	2020	2.6	18.8	19.0
Gated DPRNN [16]	Time	2020	7.5	20.1	20.4
DPTNet [17]	Time	2020	2.7	20.2	20.6
DPTCN-ATPP [18]	Time	2021	4.7	19.6	19.9
SepFormer [19]	Time	2021	26.0	20.4	20.5
Sandglasset [20]	Time	2021	2.3	20.8	21.0
Wavesplit [21]	Time	2021	29.0	21.0	21.2
TFPSNet [25]	T-F	2022	2.7	21.1	21.3
MTDS (DPTNet) [22]	Time	2022	4.0	21.5	21.7
SFSRNet [23]	Time	2022	59.0	22.0	22.1
QDPN [24]	Time	2022	200.0	22.1	-
TF-GridNet	T-F	2022	14.5	**23.5**	**23.6**

size to 1 sample, chunk size to 250 frames, and overlap between consecutive chunks to 50%, following the best configuration reported in [15]. For TF-GridNet, in each block we remove the full-band self-attention module, and set I and J to 1 (in this case, the order of LN and Unfold does not matter). From row 1 and 4, we observe that TF-GridNet with complex spectral mapping obtains better results (21.2 vs. 18.8 dB). Table II also reports the performance of using TF-GridNet with masking in row 2 and 3. In row 2, we mask learned embeddings, following [11], [15], [25]. We closely follow the encoder-masking-decoding modules used in [25], but replace their path-scanning modules with our intra-frame full-band and sub-band temporal modules. In row 3, we use TF-GridNet for complex ratio masking based separation [8], [29]. After obtaining the output tensor of the Deconv2D module (see Fig. 2), we first truncate the values in the tensor to the range $[-5, 5]$ to obtain an estimated complex ratio mask and then multiply it with the mixture spectrogram for separation. From row 2, 3 and 4, we notice that complex spectral mapping performs better (21.2 vs. 20.7 and 20.8 dB).

2) Ablation Results With Different Hyper-Parameters: Table III presents the ablation results of our models on WSJ0-2mix using different model hyper-parameters. From row 1-4, we can see that, when the kernel size is sufficiently large (i.e., $I = 8$), using the Unfold and Deconv1D mechanism together with a smaller embedding dimension (i.e., $D = 16$) does not decreases SI-SDRi, compared with the configuration that uses a larger embedding dimension (i.e., $D = 128$) but does not stack nearby T-F embeddings (i.e., $I = 1$). One benefit of using the former configuration is that the memory consumption is lower. From row 4 and 5, we can see that the MC loss produces slightly better SI-SDRi (21.6 vs. 21.8 dB). Comparing row 7 with 5 and 6, we notice that enlarging the model size by increasing the number of hidden units H in BLSTMs and the embedding dimension D produces clear improvement. The results in row 7, 8 and 9 suggest that including the full-band self-attention module is beneficial, and using four attention heads leads to better performance than just using one (22.9 vs. 22.6 dB). In row 10, we increase the embedding dimension to 48 and reduce the kernel size I to 4, and obtain slightly better SI-SDRi than the model in row 9 (23.0 vs. 22.9 dB). In row 11, we use LN+Unfold rather than Unfold+LN. This results in 0.2 dB better SI-SDRi (23.2 vs. 23.0 dB). Further enlarging model size in row 12 produces further gains (from 23.2 to 23.5 dB).

3) Comparison With Previous Models: Table IV compares the performance of our best TF-GridNet with previous models on WSJ0-2mix. Compared with previous best such as SepFormer [19], SFSRNet [23] and QDPN [24], our model has a modest size. Notice that, since 2019, T-F domain models have been largely under-explored and under-represented for anechoic speaker separation, and many research efforts have been devoted to time-domain approaches. The recent TFPSNet model [25] achieves a competitive SI-SDRi at 21.1 dB, but the performance still falls within the range of scores (i.e., [20.0, 22.0] dB SI-SDRi) that can be commonly achieved by modern time-domain models. Our study, for the first time since 2019, unveils that complex T-F domain models, with a contemporary DNN architecture, can outperform modern time-domain models by a large margin. Later in Section VI-E, we will provide the computational cost of TF-GridNet.

TABLE V
RESULTS ON SMS-WSJ (1CH)

Systems	Δ_l	Δ_r	Loss	SI-SDR (dB)	SDR (dB)	PESQ	eSTOI	WER (%)
Unprocessed	-	-	-	−5.5	−0.4	1.50	0.441	78.40
DNN_1	-	-	(10)	16.2	17.2	3.45	0.924	9.49
DNN_1	-	-	(11)	14.7	15.7	3.35	0.914	9.64
DNN_1	-	-	(12)	15.7	16.6	3.41	0.924	9.26
DNN_1+DNN_2	-	-	(12)	16.6	17.6	3.53	0.934	8.80
DNN_1+SCMFWF+DNN_2	39	0	(12)	17.6	18.7	3.67	0.946	8.51
DNN_1+SCMFWF+DNN_2	20	19	(12)	**18.4**	**19.6**	**3.70**	**0.952**	**7.91**
DNN_1+WPE+DNN_2	40	-	(12)	17.5	18.6	3.67	0.947	8.19
DPRNN-TasNet [15]	-	-	-	6.5	-	2.28	0.734	38.10
$SISO_1$ [35]	-	-	-	5.7	-	2.40	0.748	28.70
DNN_1+(FCP+DNN_2)×2 [35]	-	-	-	12.7	14.1	3.25	0.899	12.80
DNN_1+(msFCP+DNN_2)×2 [39]	-	-	-	13.4	-	3.41	-	10.90
Oracle direct-path signal	-	-	-	∞	∞	4.50	1.000	6.28

TABLE VI
RESULTS ON SMS-WSJ (2CH)

Systems	Δ_l	Δ_r	Loss	SI-SDR (dB)	SDR (dB)	PESQ	eSTOI	WER (%)
Unprocessed	-	-	-	−5.5	−0.4	1.50	0.441	78.40
DNN_1	-	-	(10)	17.8	19.1	3.67	0.946	8.33
DNN_1	-	-	(11)	16.0	17.3	3.52	0.936	8.31
DNN_1	-	-	(12)	17.7	18.9	3.68	0.950	7.68
DNN_1+DNN_2	-	-	(12)	17.7	18.9	3.68	0.949	7.90
DNN_1+MCMFWF+DNN_2	0	0	(12)	17.8	19.0	3.70	0.950	7.84
DNN_1+MCMFWF+DNN_2	29	0	(12)	19.9	21.5	3.79	0.965	**7.12**
DNN_1+MCMFWF+DNN_2	15	14	(12)	**20.3**	**22.0**	**3.81**	**0.967**	7.41
DNN_1+ConvBF+DNN_2	29	-	(12)	19.4	20.9	3.80	0.961	7.52
MC-ConvTasNet [79]	-	-	-	5.8	-	2.16	0.720	45.70
FasNet+TAC [80]	-	-	-	6.9	-	2.27	0.731	34.80
$MISO_1$ [35]	-	-	-	8.2	-	2.85	0.826	17.20
$MISO_1$-BF-$MISO_3$ [35]	-	-	-	12.7	-	3.43	0.907	10.70
DNN_1+(msFCP$_{MVDR}$ +msFCP+DNN_2)×2 [39]	-	-	-	15.8	-	3.71	-	8.60
Oracle direct-path signal	-	-	-	∞	∞	4.50	1.000	6.28

TABLE VII
RESULTS ON SMS-WSJ (6CH)

Systems	Δ_l	Δ_r	Loss	SI-SDR (dB)	SDR (dB)	PESQ	eSTOI	WER (%)
Unprocessed	-	-	-	−5.5	−0.4	1.50	0.441	78.40
DNN_1	-	-	(10)	19.6	21.0	3.87	0.961	7.63
DNN_1	-	-	(11)	19.4	20.8	3.83	0.964	6.92
DNN_1	-	-	(12)	19.9	21.2	3.89	0.966	7.27
DNN_1+DNN_2	-	-	(12)	19.9	21.2	3.89	0.966	7.34
DNN_1+MCMFWF+DNN_2	0	0	(12)	20.1	21.4	3.90	0.967	7.28
DNN_1+MCMFWF+DNN_2	9	0	(12)	22.6	24.6	4.04	0.978	**6.65**
DNN_1+MCMFWF+DNN_2	5	4	(12)	**22.8**	**24.9**	**4.08**	**0.980**	6.76
DNN_1+ConvBF+DNN_2	9	-	(12)	21.9	23.7	4.00	0.975	6.74
FasNet+TAC [81]	-	-	-	8.6	-	2.37	0.771	29.80
MC-ConvTasNet [80]	-	-	-	10.8	-	2.78	0.844	23.10
$MISO_1$ [35]	-	-	-	10.2	-	3.05	0.859	14.00
LBT [82]	-	-	-	13.2	14.8	3.33	0.910	9.60
$MISO_1$-BF-$MISO_3$ [35]	-	-	-	15.6	-	3.76	0.942	8.30
Oracle direct-path signal	-	-	-	∞	∞	4.50	1.000	6.28

TABLE VIII
RESULTS ON WHAMR! (1CH)

Systems	Δ_l	Δ_r	Loss	SI-SDR (dB)	SDR (dB)	PESQ	eSTOI
Unprocessed	-	-	-	−6.1	−3.5	1.41	0.317
DNN_1	-	-	(10)	11.0	12.1	2.69	0.790
DNN_1	-	-	(11)	10.3	11.4	2.72	0.787
DNN_1	-	-	(12)	10.6	11.7	2.75	0.793
DNN_1+DNN_2	-	-	(12)	10.7	11.8	2.74	0.794
DNN_1+SCMFWF+DNN_2	20	19	(12)	**11.2**	**12.3**	**2.79**	**0.808**
Conv-TasNet [11], [42]	-	-	-	2.2	-	-	-
$SISO_1$ [35]	-	-	-	4.2	6.2	1.79	0.594
3-Stage BLSTM-TasNet [42]	-	-	-	4.8	-	-	-
Wavesplit [21]	-	-	-	5.9	-	-	-
Gated DPRNN [16]	-	-	-	6.1	-	-	-
QDPN [24]	-	-	-	7.0	-	-	-
DNN_1+(FCP+DNN_2)×2 [37]	-	-	-	7.4	8.9	2.39	0.743
Wavesplit + DM [21]	-	-	-	7.1	8.7	-	-
SUDO RM -RF + DM [14]	-	-	-	7.4	-	-	-
SepFormer + DM [19], [83]	-	-	-	7.9	9.5	-	-
QDPN + DM [24]	-	-	-	8.3	-	-	-

TABLE IX
RESULTS ON WHAMR! (2CH)

Systems	Δ_l	Δ_r	Loss	SI-SDR (dB)	SDR (dB)	PESQ	eSTOI
Unprocessed	-	-	-	−6.1	−3.5	1.41	0.317
DNN_1	-	-	(10)	12.8	14.0	3.00	0.844
DNN_1	-	-	(11)	12.0	13.2	3.01	0.839
DNN_1	-	-	(12)	12.5	13.5	3.05	0.846
DNN_1+DNN_2	-	-	(12)	12.5	13.6	3.05	0.846
DNN_1+MCMFWF+DNN_2	15	14	(12)	**13.7**	**14.8**	**3.16**	**0.868**
MC-ConvTasNet [80], [84]	-	-	-	6.0	-	-	-
MC-ConvTasNet with speaker extraction [84]	-	-	-	7.3	-	-	-

B. Results on SMS-WSJ and WHAMR!

This section evaluates TF-GridNet and the two-DNN system on reverberant and noisy-reverberant speaker separation. In the following experiments, in default we use $B = 4$ and $H = 192$ to save computation[7] and use LN+Unfold. Based on the validation sets, we set $I = 4$, $J = 1$ and $D = 48$ for SMS-WSJ, and $I = 8$, $J = 1$ and $D = 24$ for WHAMR!.

1) Comparison of Loss Functions: The speaker separation community usually uses SI-SDR as the key evaluation metric and many previous models are trained to optimize SI-SDR. We also do this in our experiments on WSJ0-2mix in order to compare TF-GridNet with earlier models. However, using SI-SDR as the loss is known to produce sub-optimal magnitude estimates due to the compensation between estimated magnitude and phase [63], while metrics such as PESQ, eSTOI and WER favor signals with good magnitude estimates. Based on SMS-WSJ and WHAMR!, in Tables V, VI, VII, VIII and IX we make a direct comparison of training TF-GridNet (i.e., DNN_1) with the SI-SDR+MC loss in (10), Wav+Mag in (11) and Wav+Mag+MC in (12). We observe that, compared with SI-SDR+MC, Wav+Mag+MC performs better or comparably good in PESQ, eSTOI and WER, and slightly worse in SI-SDR and SDR; and compared with Wav+Mag, it usually performs better. We will in default use the Wav+Mag+MC loss in the following experiments.

2) Comparison in Monaural, Single-DNN Setup: Tables V and VIII respectively present the results of TF-GridNet (denoted as DNN_1) on the monaural tasks of SMS-WSJ and WHAMR!. TF-GridNet substantially outperforms competing systems that train a single DNN for separation. For example, in Table V TF-GridNet is 9.2 dB better than DPRNN-TasNet (15.7 vs. 6.5 dB

[7]We also experimented with larger TF-GridNets and observed better performance but we consider this unnecessary. We will show later that TF-GridNet with this setup already produces better results than competing models.

SI-SDR) [15] and 10.0 dB better than TCN-DenseUNet based SISO_1 (15.7 vs. 5.7 dB SI-SDR) [35]. To obtain state-of-the-art performance, many previous speaker separation studies tend to use dynamic mixing (DM) to generate more training mixtures. Their DM results on the monaural task of WHAMR! are listed in the bottom panel of Table VIII. Although DM yields slight improvements for previous models, their final performance is still worse than the 10.6 dB SI-SDR result obtained by TF-GridNet without DM (i.e., the DNN_1 row). These results show the effectiveness of TF-GridNet for noisy-reverberant speaker separation.

3) Comparison in Multi-Channel, Single-DNN Setup: Tables VI and VII respectively present the results of TF-GrdiNet based DNN_1 for two- and six-channel separation on SMS-WSJ, and Table IX reports two-channel results on WHAMR!. TF-GridNet shows substantially better performance than competing single-DNN approaches. For example, in Table VII TF-GridNet obtains 19.9 dB SI-SDR, while FasNet+TAC [81], MC-Conv-TasNet [80], TCN-DenseUNet [35] and LBT [82] respectively obtain 8.6, 10.8, 10.2 and 13.2 dB.

4) Effectiveness of Including MFWF and Post-Filtering: For the post-filtering network (i.e., DNN_2), which is trained in an enhancement way, we use the same configuration as DNN_1 but use $B = 3$ TF-GridNet blocks. Although TF-GridNet based DNN_1 already exhibits strong separation performance, we observe that using its outputs to compute an MFWF and another TF-GridNet for post-filtering still produces clear improvements. This can be observed in Tables VI and VII by comparing DNN_1+MCMFWF+DNN_2, DNN_1, and DNN_1+DNN_2 (which stacks two TF-GridNets but not performing linear filtering in between). In the monaural case, in Table V DNN_1+SCMFWF+DNN_2 is also better than DNN_1.

5) MFWF vs. Other Linear Filters: In Tables VI and VII, we observe that using MCMFWF with both past and future context (i.e., $\Delta_l > 0$ and $\Delta_r > 0$) between DNN_1 and DNN_2 produces clear improvements over MCMFWF with only past context (i.e., $\Delta_l > 0$ and $\Delta_r = 0$), MCMFWF with no context (i.e., $\Delta_l = 0$ and $\Delta_r = 0$), and convolutional beamformer. In Table V, SCMFWF with both past and future context leads to better scores than WPE as well as SCMFWF with only past context in the single-channel case.

C. Results on WSJ0CAM-DEREVERB

For the rest experiments (including the ones in this subsection and in the next subsection), we set $I = 4$, $J = 2$, and $D = 48$. J is increased to 2 as the sampling rate is 16 kHz. The other setups are the same as those in the previous subsection.

Tables X and XI respectively present the results of using TF-GridNet for one- and eight-channel dereverberation. Trained to perform complex spectral mapping, DNN_1 based on TF-GridNet achieves substantially better performance than SISO_1 (16.6 vs. 8.4 dB SI-SDR in Table X) and MISO_1 (19.9 vs. 11.3 dB SI-SDR in Table XI) proposed in [36], which also uses complex spectral mapping but with TCN-DenseNet. With beamforming and post-filtering, DNN_1+MCMFWF+DNN_2 based on TF-GridNet also shows better results than the competing approach (21.2 vs. 18.2 dB SI-SDR) in [36], which uses two TCN-DenseUNets with a composition of linear filters.

TABLE X
RESULTS ON WSJ0CAM-DEREVERB (1CH)

Systems	Δ_l	Δ_r	Loss	SI-SDR (dB)	PESQ	eSTOI
Unprocessed	-	-	-	−3.6	1.64	0.494
DNN_1	-	-	(11)	16.6	3.72	0.947
DNN_1+DNN_2	-	-	(11)	17.0	3.77	0.948
DNN_1+SCMFWF+DNN_2	20	19	(11)	**17.3**	**3.78**	**0.950**
SISO_1 [36]	-	-	-	8.4	3.12	0.868
DNN_1+(FCP+DNN_2)×2 [37]	-	-	-	12.7	3.46	-
SISO_1+FCP_{WPE}+WPE+SISO_5 [36]	-	-	-	12.7	3.49	0.919

TABLE XI
RESULTS ON WSJ0CAM-DEREVERB (8CH)

Systems	Δ_l	Δ_r	Loss	SI-SDR (dB)	PESQ	eSTOI
Unprocessed	-	-	-	−3.6	1.64	0.494
DNN_1	-	-	(11)	19.9	3.95	0.971
DNN_1+DNN_2	-	-	(11)	20.3	4.00	0.972
DNN_1+MCMFWF+DNN_2	4	3	(11)	**21.2**	**4.02**	**0.975**
MISO_1 [36]	-	-	-	11.3	3.49	0.921
MISO_1+$\text{FCP}_{\text{mWMPDR}_{\text{WPE}}}$+ mWMPDR$_{\text{WPE}}$+WPE+$\text{MISO}_{10}$ [36]	-	-	-	18.2	3.98	0.967

TABLE XII
RESULTS ON L3DAS22 3D SPEECH ENHANCEMENT TASK (8CH)

Systems	Δ_l/Δ_r	#params (M)	Loss	WER (%)	STOI	Task1Metric
DNN_1	-	5.6	(26)	1.68	0.988	0.985
DNN_1+DNN_2	-	9.8	(26)	1.63	0.989	0.986
DNN_1+MCMFWF+DNN_2	4/3	9.8	(26)	**1.29**	**0.994**	**0.990**
Winner (ESP-SE) [43]	-	13.8	-	1.89	0.987	0.984
Runner-up (BaiduSpeech) [85]	-	-	-	2.50	0.975	0.975
3nd-place (PCG-AIID) [87]	-	-	-	3.20	0.972	0.970
Challenge baseline [88]	-	5.5	-	21.20	0.878	0.833

From the last two rows of Table XI, we notice that, based on TCN-DenseUNet, using complicated sub-band linear filtering followed by post-filtering (i.e., the last row) produces large improvement over MISO_1 (18.2 vs. 11.3 dB SI-SDR) [36]. This indicates that the sub-band linear filters can model what TCN-DenseUNet, which performs full-band modeling, is not good at modeling. In comparison, using a single TF-GridNet alone is already better than the last two rows (i.e., 19.9 vs. 11.3 and 18.2 dB SI-SDR) and the improvement brought by beamforming and post-filtering is not large (21.2 vs. 19.9 dB SI-SDR). This indicates that TF-GridNet could, to a large extent, model what sub-band linear filters complement to full-band models, likely through the sub-band temporal modules.

D. Results on L3DAS22

L3DAS22 requires participants to estimate the dry source signal. Following (11), we define the loss as

$$\mathcal{L}_{\text{Wav+Mag,GEQ}} = \sum_{c=1}^{C} \left(\frac{1}{N} \|\hat{\alpha}^{(c)}\hat{o}(c) - o(c)\|_1 + \frac{1}{T \times F} \left\| \left|\text{STFT}(\hat{\alpha}^{(c)}\hat{o}(c))\right| - \left|\text{STFT}(o(c))\right| \right\|_1 \right), \quad (26)$$

TABLE XIII
ABLATION RESULTS OF COMPUTATION COST AND SEPARATION PERFORMANCE ON WSJ0-2MIX

Row	Systems	Unfold+LN or LN+Unfold	$L/D/I/J/H$	#params (M)	Loss	SI-SDRi (dB)	Window/hop sizes (ms)	GMAC/s	Memory cost (GPU)		Speed (GPU)			Speed (CPU)
									Forward (MB/seg)	Backward (MB/seg)	Training (min/epoch)	Forward (ms/seg)	Backward (ms/seg)	Forward (ms/seg)
1	TF-GridNet	LN+Unfold	4/64/4/1/256	14.5	(10)	23.5	32/8	231.1	4918.8	13 663.2	151.4	450.8	832.9	18 440.6
2	TF-GridNet	LN+Unfold	4/48/4/1/192	8.2	(10)	23.2	32/8	131.1	4581.9	10 871.4	119.5	1439.0	609.7	12 169.2
3	TF-GridNet	LN+Unfold	4/48/4/1/192	8.2	(10)	23.2	16/8	66.0	2520.1	5803.3	71.6	756.5	404.0	6383.2
4	TF-GridNet	LN+Unfold	4/96/2/2/192	8.4	(10)	22.2	16/8	36.2	2282.6	3853.7	24.2	364.0	188.8	3564.0
5	TF-GridNet	LN+Unfold	4/64/3/3/192	8.2	(10)	21.3	16/8	24.4	1620.6	2686.9	18.0	247.2	132.1	2428.9
6	TF-GridNet	LN+Unfold	4/48/4/4/192	8.2	(10)	20.6	16/8	19.2	1318.7	2189.4	15.3	193.0	106.4	1930.8
7	TF-GridNet	LN+Unfold	4/32/4/4/128	3.7	(10)	20.0	16/8	9.5	956.3	1590.8	12.6	27.8	58.0	1206.7
8	TF-GridNet	LN+Unfold	4/24/4/4/ 96	2.1	(10)	18.9	16/8	6.1	785.9	1300.0	11.4	17.0	62.1	903.7
9	TF-GridNet	LN+Unfold	4/88/2/2/172	6.8	(10)	22.0	16/8	29.8	2093.2	3496.0	23.1	325.5	192.6	3190.0
10	Conv-TasNet [11]	-	-	5.1	-	15.6	2/1	5.1	1326.8	1445.2	7.4	88.1	26.0	602.2
11	DPRNN [15]	-	-	2.6	-	18.8	0.25/0.125	42.2	2927.4	7049.6	28.6	121.8	228.6	4109.4
12	SepFormer [19]	-	-	26.0	-	20.4	2/1	59.5	5465.7	5703.9	N/A	82.2	135.0	5734.1
13	TFPSNet [25]	-	-	2.7	-	21.1	32/16	29.6	4869.4	7182.0	35.2	77.4	184.5	4486.8

Notes:
(a) TF-GridNet is configured to always include the self-attention module.
(b) GMAC/s is computed based on a 4-second segment and batch size 1.
(c) For the memory cost on GPU in the forward/backward pass, we report it in megabytes (MB) per 4-second segment.
(d) For the training speed, we report the time in minutes to finish an epoch with 20,000 4-second segments on an NVIDIA A100 GPU with 40 GB memory. The batch size is set so that nearly all the GPU memory is utilized. The result of SepFormer is not available, because, in each training step, it is designed to model an entire mixture.
(e) For the forward (and backward) speed on GPU, we report the averaged time in millisecond (ms) taken to finish processing 4-second segments with a batch size of 1. An NVIDIA Tesla V100 GPU with 32 GB memory is used.
(f) For the forward speed on CPU, we also report the averaged time in millisecond to process 4-second segments with a batch of 1. A single core of a CPU, Intel(R) Xeon(R) Gold 6242 CPU @ 2.80GHz, is used.
(g) For the columns on forward/backward memory and speed, we use Pytorch v2.0.1 and *torch.profiler* for profiling.

where $o(c)$ denotes the dry source signal of speaker c (see our physical model in (1)) and $\hat{\alpha}^{(c)} = \text{argmin}_\alpha \|\alpha\hat{o}^{(c)} - o^{(c)}\|_2^2 = (\hat{o}^{(c)})^\mathsf{T} o^{(c)} / (\hat{o}^{(c)})^\mathsf{T}\hat{o}^{(c)}$ is a gain equalization (GEQ) factor [40], [43] that allows estimated speech to have an energy level different from target speech.

Table XII reports the results. A single TF-GridNet (i.e., DNN_1) already outperforms our winning solution [43] and the rest 16 submissions (see this link[8] for the challenge ranking), including the runner-up system [85], whose monaural version [86] won the recent DNS2022 and AEC2022 challenges.

Including beamforming and post-filtering yields further improvement. Here MCMFWF is computed in a way similarly to (13), but we project the far-field B-format Ambisonic mixture to the dry source signal estimated by DNN_1 so that the beamforming result can be potentially time-aligned with the dry target, if the dry target estimated by DNN_1 is reasonably good, which is the case from the DNN_1 row. In comparison, DNN-supported convolutional beamformer cannot produce an estimate time-aligned with the dry source, and how to modify it to deal with B-format Ambisonic signals is unknown.

E. Computation Cost vs. Separation Performance

Although this article focuses on the separation performance rather than computation cost, this subsection varies the computation cost of TF-GridNet and reports the separation performance on WSJ0-2mix in Table XIII (see the notes below the table for how we calculate the computation cost). The computation cost can be controlled by increasing the stride size J (so that the sequence length is reduced), reducing the overlap between consecutive frames, and reducing the hidden dimensions of BLSTMs, H, as well as the embedding dimension D.

In row 1, the model obtaining the 23.5 dB result (i.e., the best result in Table IV) is very costly, requiring 231.1 GMAC/s which is quite high. In row 2, we reduce the hidden units in BLSTMs, H, from 256 to 196 and the embedding dimension D from 64 to 48. This reduces GMAC/s to 131.1, while the performance degrades slightly to 23.2 dB. In row 3, we decrease the window size from 32 to 16 ms, reducing GMAC/s by almost half to 66.0, as the number of frequencies is cut by around half. The performance, nonetheless, remains at 23.2 dB. In row 4–6, we increase the stride size J from 1 to 2, 3 and 4 respectively, set the kernel size I equal to J, and set D such that $D \times I = H$. These reduce GMAC/s from 66.0 gradually down to 19.2, as the sequence length the BLSTMs need to model becomes shorter. In row 7, we reduce the hidden units in BLSTMs, H, from 192 to 128. The computation is further reduced to 9.5 GMAC/s, and the performance is at 20.0 dB. In row 8, we reduce the hidden units in BLSTMs, H, from 128 to 96, resulting in 6.1 GMAC/s. The model can still obtain 18.9 dB.

Table XIII provides the training speed on a modern GPU in terms of the number of minutes taken to finish an epoch. Although the models in the first two rows take a long time (i.e., 119.5 and 231.1 minutes) to complete an epoch, all the other configurations have a reasonable training time per epoch.

Table XIII compares the computation cost of several other representative models with that of TF-GridNet. TF-GridNet obtains competitive performance, given limited computation cost. For example, TF-GridNet in row 9 obtains better separation performance than TFPSNet [25] in row 13 (i.e, 22.0 vs. 21.1 dB SI-SDRi), using similar GMAC/s (i.e., 29.8 vs. 29.6), less memory, and exhibiting faster inference speed on CPU. Compared with a representative time-domain model, DPRNN [15], shown in row 11, the TF-GridNets from row 4 to 9 can all obtain better separation performance using fewer GMAC/s.

These results suggest that TF-GridNet can be configured, in a flexible way, to use a reasonable amount of computation and achieve a reasonable separation performance.

[8]See https://www.l3das.com/icassp2022/results.html

VII. Conclusion

We have proposed TF-GridNet, a DNN architecture modeling complex T-F spectrograms, for single- and multi-channel speech separation. By integrating full- and sub-band modeling inside TF-GridNet and outside through beamforming and post-filtering, the proposed systems achieve state-of-the-art performance for speech separation in noisy-reverberant conditions on multiple public datasets. Our future research will extend TF-GridNet for real-time, online speech separation, building upon our preliminary investigations [79], [89], [90] which have shown promising results.

TF-GridNet obtains a state-of-the-art 23.5 dB SI-SDRi on WSJ0-2mix, and it can be configured to use a reasonable amount of computation and achieve a reasonable separation performance. These resulte highlights the strong performance of T-F domain models also for anechoic speaker separation, suggesting that T-F domain methods modeling complex representations, which implicitly perform phase estimation by predicting target RI components simultaneously, are not sub-optimal compared to time-domain approaches for the task of anechoic speaker separation. The performance differences between these two approaches observed in earlier studies could mainly result from their differences in DNN architectures.

In closing, we emphasize that (i) the patterns of speech spectrograms vary with frequency but, within each sub-band, some patterns such as spatial and reverberation patterns are relatively stable along time; and (ii) full-band or sub-band modeling alone is likely not capable of sufficiently modeling such patterns. Our proposed ways to integrate them exhibit excellent performance in our experiments. The meta-idea of integrated full- and sub-band modeling, we believe, would motivate the design of many new algorithms in future research on neural speech separation.

Acknowledgment

We would like to thank Dr. Wangyou Zhang at SJTU for generously sharing his reproduced code of TFPSNet.

References

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

[2] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2016, pp. 31–35.

[3] 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.

[4] Y. Isik et al., "Single-channel multi-speaker separation using deep clustering," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2016, pp. 545–549.

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

[6] Z.-Q. Wang, J. Le 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.

[7] 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.

[8] Y. Liu and D. Wang, "Divide and conquer: A deep CASA approach to talker-independent monaural speaker separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 12, pp. 2092–2102, Dec. 2019.

[9] 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.

[10] 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.

[11] 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.

[12] M. W. Lam, J. Wang, D. Su, and D. Yu, "Mixup-breakdown: A consistency training method for improving generalization of speech separation models," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 6374–6378.

[13] L. Zhang et al., "FurcaNeXt: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks," in *Proc. Int. Conf. Multimedia Model.*, 2020, pp. 653–665.

[14] E. Tzinis, Z. Wang, and P. Smaragdis, "Sudo RM -RF: Efficient networks for universal audio source separation," in *Proc. IEEE 30th Int. Workshop Mach. Learn. Signal Process.*, 2020, pp. 1–6.

[15] 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.

[16] E. Nachmani et al., "Voice separation with an unknown number of multiple speakers," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 7121–7132.

[17] 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. 2642–2646.

[18] Y. Zhu, X. Zheng, X. Wu, W. Liu, L. Pi, and M. Chen, "DPTCN-ATPP: Multi-scale end-to-end modeling for single-channel speech separation," in *Proc. IEEE 5th Int. Conf. Commun. Inf. Syst.*, 2021, pp. 39–44.

[19] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 21–25.

[20] M. W. Y. Lam, J. Wang, D. Su, and D. Yu, "Sandglasset: A light multi-granularity self-attentive network for time-domain speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 5759–5763.

[21] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2840–2849, 2021.

[22] S. Qian, L. Gao, H. Jia, and Q. Mao, "Efficient monaural speech separation with multiscale time-delay sampling," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 6847–6851.

[23] J. Rixen and M. Renz, "SFSRNet: Super-resolution for single-channel audio source separation," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 11220–11228.

[24] J. Rixen and M. Renz, "QDPN - Quasi-dual-path network for single-channel speech separation," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 5353–5357.

[25] L. Yang, W. Liu, and W. Wang, "TFPSNet: Time-frequency domain path scanning network for speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 6842–6846.

[26] X. Le, H. Chen, K. Chen, and J. Lu, "DPCRN: Dual-path convolution recurrent network for single channel speech enhancement," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 821–825.

[27] F. Dang, H. Chen, and P. Zhang, "DPT-FSNet:Dual-Path transformer based full-band and sub-band fusion network for speech enhancement," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 6857–6861.

[28] Z.-Q. Wang et al., "TF-GridNet: Making time-frequency domain models great again for monaural speaker separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023.

[29] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 3, pp. 483–492, Mar. 2016.

[30] S.-W. Fu, T.-Y. Hu, Y. Tsao, and X. Lu, "Complex spectrogram enhancement by convolutional neural network with multi-metrics learning," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, 2017, pp. 1–6.

[31] K. Tan and D. Wang, "Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 380–390, 2020.

[32] Z.-Q. Wang and D. Wang, "Deep learning based target cancellation for speech dereverberation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 941–950, 2020.

[33] Z.-Q. Wang, P. Wang, and D. Wang, "Complex spectral mapping for single- and multi-channel speech enhancement and robust ASR," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1778–1787, 2020.

[34] Z.-Q. Wang and D. Wang, "Multi-microphone complex spectral mapping for speech dereverberation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 486–490.

[35] Z.-Q. Wang, P. Wang, and D. Wang, "Multi-microphone complex spectral mapping for utterance-wise and continuous speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2001–2014, 2021.

[36] Z.-Q. Wang, G. Wichern, and J. Le Roux, "Leveraging low-distortion target estimates for improved speech enhancement," 2021, *arXiv:2110.00570*.

[37] Z.-Q. Wang, G. Wichern, and J. L. Roux, "Convolutive prediction for monaural speech dereverberation and noisy-reverberant speaker separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3476–3490, 2021.

[38] K. Tan, Z.-Q. Wang, and D. Wang, "Neural spectrospatial filtering," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 605–621, 2022.

[39] Z.-Q. Wang, G. Wichern, and J. Le Roux, "Convolutive prediction for reverberant speech separation," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2021, pp. 56–60.

[40] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR - Half-baked or well done?," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 626–630.

[41] L. Drude et al., "SMS-WSJ: Database, performance measures, and baseline recipe for multi-channel source separation and recognition," 2019, *arXiv:1910.13934*.

[42] M. MacIejewski, G. Wichern, E. McQuinn, and J. L. Roux, "WHAMR!: Noisy and reverberant single-channel speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 696–700.

[43] Y.-J. Lu et al., "Towards low-distortion multi-channel speech enhancement: The ESPNet-SE submission to the L3DAS22 challenge," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9201–9205.

[44] E. Guizzo et al., "L3DAS22 challenge: Learning 3D audio sources in a real office environment," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9186–9190.

[45] Y.-J. Lu et al., "ESPnet-SE: Speech enhancement for robust speech recognition, translation, and understanding," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 5458–5462.

[46] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[47] Z.-Q. Wang and D. Wang, "Combining spectral and spatial features for deep learning based blind speaker separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 2, pp. 457–468, Feb. 2019.

[48] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multi-microphone speech enhancement and source separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 4, pp. 692–730, Apr. 2017.

[49] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B. -H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 7, pp. 1717–1731, Sep. 2010.

[50] R. Haeb-Umbach, J. Heymann, L. Drude, S. Watanabe, M. Delcroix, and T. Nakatani, "Far-field automatic speech recognition," in *Proc. IEEE*, vol. 109, no. 2, pp. 124–148, Feb. 2021.

[51] A. Courville et al., *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[52] E. A. P. Habets and P. A. Naylor, "Dereverberation," in *Audio Source Separation and Speech Enhancement*, E. Vincent, T. Virtanen, and S. Gannot, Eds. Hoboken, NJ, USA: Wiley, 2018, pp. 317–343.

[53] T. Nakatani and K. Kinoshita, "A unified convolutional beamformer for simultaneous denoising and dereverberation," *IEEE Signal Process. Lett.*, vol. 26, no. 6, pp. 903–907, Jun. 2019.

[54] Y. Zhao, D. Wang, B. Xu, and T. Zhang, "Late reverberation suppression using recurrent neural networks with long short-term memory," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 5434–5438.

[55] R. Zhou, W. Zhu, and X. Li, "Single-channel speech dereverberation using subband network with a reverberation time shortening target," 2022, *arXiv:2204.08765*.

[56] C. Quan and X. Li, "Multichannel speech separation with narrow-band conformer," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 5378–5382.

[57] X. Hao, X. Su, R. Horaud, and X. Li, "FullSubNet: A full-band and sub-band fusion model for real-time single-channel speech enhancement," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 6633–6637.

[58] J. Chen, Z. Wang, D. Tuo, Z. Wu, S. Kang, and H. Meng, "FullSubNet: Channel attention FullSubNet with complex spectrograms for speech enhancement," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 7857–7861.

[59] C. Xu, W. Rao, X. Xiao, E. S. Chng, and H. Li, "Single channel speech separation with constrained utterance level permutation invariant training using grid LSTM," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 6–10.

[60] Y. Liu, B. Thoshkahna, A. Milani, and T. Kristjansson, "Voice and accompaniment separation in music using self-attention convolutional neural network," 2020, *arXiv:2003.08954*.

[61] A. Pandey and D. Wang, "Dense CNN with self-attention for time-domain speech enhancement," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1270–1279, 2021.

[62] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[63] Z.-Q. Wang, G. Wichern, and J. Le Roux, "On the compensation between magnitude and phase in speech separation," *IEEE Signal Process. Lett.*, vol. 28, pp. 2018–2022, 2021.

[64] 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.

[65] S. Wisdom et al., "Differentiable consistency constraints for improved deep speech enhancement," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 900–904.

[66] K. Zmolikova and J. H. Cernock, "BUT system for the first clarity enhancement challenge," in *Proc. Clarity*, 2021, pp. 1–3.

[67] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, Sep. 2011.

[68] Z.-Q. Wang et al., "Sequential multi-frame neural beamforming for speech separation and enhancement," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 905–911.

[69] Z.-Q. Wang, G. Wichern, S. Watanabe, and J. Le Roux, "STFT-Domain neural speech enhancement with very low algorithmic latency," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 397–410, 2023.

[70] K. Kinoshita, M. Delcroix, H. Kwon, T. Mori, and T. Nakatani, "Neural network-based spectrum estimation for online WPE dereverberation," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 384–388.

[71] S. Cornell, M. Pariente, F. Grondin, and S. Squartini, "Learning filterbanks for end-to-end acoustic beamforming," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 6507–6511.

[72] K. Kinoshita et al., "A summary of the REVERB challenge: State-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP J. Adv. Signal Process.*, vol. 2016, no. 1, pp. 1–19, 2016.

[73] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 351–355.

[74] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 829–852, 2022.

[75] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A Python package for weighted prediction error dereverberation in numpy and tensorflow for online and offline processing," in *Proc. IEEE 13th Speech Commun. ITG-Symp.*, 2018, pp. 1–5.

[76] T. Yoshioka et al., "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2015, pp. 436–443.

[77] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, "BLSTM supported GEV beamformer front-end for the 3rd CHiME challenge," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2015, pp. 444–451.

[78] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

[79] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, "Neural speech enhancement with very low algorithmic latency and complexity via integrated full- and sub-band modeling," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.

[80] J. Zhang, C. Zorila, R. Doddipatla, and J. Barker, "On end-to-end multi-channel time domain speech separation in reverberant environments," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 6389–6393.

[81] Y. Luo, Z. Chen, N. Mesgarani, and T. Yoshioka, "End-to-end microphone permutation and number invariant multi-channel speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 6394–6398.

[82] H. Taherian, K. Tan, and D. Wang, "Multi-channel talker-independent speaker separation through location-based training," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 2791–2800, 2022.

[83] C. Subakan et al., "On using transformers for speech-separation," 2022, *arXiv:2202.02884*.

[84] J. Zhang, C. Zorilă, R. Doddipatla, and J. Barker, "Time-domain speech extraction with spatial information and multi speaker conditioning mechanism," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 6084–6088.

[85] G. Zhang, C. Wang, L. Yu, and J. Wei, "Multi-scale temporal frequency convolutional network with axial attention for multi-channel speech enhancement," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9206–9210.

[86] G. Zhang, L. Yu, C. Wang, and J. Wei, "Multi-scale temporal frequency convolutional network with axial attention for speech enhancement," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9122–9126.

[87] J. Li, Y. Zhu, D. Luo, Y. Liu, G. Cui, and Z. Li, "The PCG-AIID system for L3DAS22 challenge: MIMO and MISO convolutional recurrent network for multi channel speech enhancement and speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9211–9215.

[88] X. Ren et al., "A neural beamforming network for B-format 3D speech enhancement and recognition," in *Proc. IEEE 31st Int. Workshop Mach. Learn. Signal Process.*, 2021, pp. 1–6.

[89] S. Cornell, Z.-Q. Wang, Y. Masuyama, S. Watanabe, M. Pariente, and N. Ono, "Multi-channel target speaker extraction with refinement: The WAVLAB submission to the second clarity enhancement challenge," in *Proc. Clarity*, 2022, pp. 1–3.

[90] S. Cornell et al., "Multi-channel speaker extraction with adversarial training: The WAVLAB submission to the clarity ICASSP 2023 grand challenge," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–2.