TEA-PSE 2.0: SUB-BAND NETWORK FOR REAL-TIME PERSONALIZED SPEECH ENHANCEMENT

Yukai Ju[1,2,†], *Shimin Zhang*[1], *Wei Rao*[2], *Yannan Wang*[2], *Tao Yu*[2], *Lei Xie*[1,*], *Shidong Shang*[2]

[1] Audio, Speech and Language Processing Group (ASLP@NPU),
Northwestern Polytechnical University, Xi'an, China
[2] Tencent Ethereal Audio Lab, Tencent Corporation, Shenzhen, China

ABSTRACT

Personalized speech enhancement (PSE) utilizes additional cues like speaker embeddings to remove background noise and interfering speech and extract the speech from target speaker. Previous work, the Tencent-Ethereal-Audio-Lab personalized speech enhancement (TEA-PSE) system, ranked 1st in the ICASSP 2022 deep noise suppression (DNS2022) challenge. In this paper, we expand TEA-PSE to its sub-band version – TEA-PSE 2.0, to reduce computational complexity as well as further improve performance. Specifically, we adopt finite impulse response filter banks and spectrum splitting to reduce computational complexity. We introduce a time frequency convolution module (TFCM) to the system for increasing the receptive field with small convolution kernels. Besides, we explore several training strategies to optimize the two-stage network and investigate various loss functions in the PSE task. TEA-PSE 2.0 significantly outperforms TEA-PSE in both speech enhancement performance and computation complexity. Experimental results on the DNS2022 blind test set show that TEA-PSE 2.0 brings 0.102 OVRL personalized DNSMOS improvement with only 21.9% multiply-accumulate operations compared with the previous TEA-PSE.

Index Terms— personalized speech enhancement, sub-band, real-time, deep learning

1. INTRODUCTION

Real-time communication (RTC) becomes indispensable in our daily life. However, the speech quality is significantly affected by background noise, reverberation, speech from background speakers, etc. Effective speech enhancement plays an important role in the RTC system. Conventional speech enhancement mainly focuses on removing the background noise and reverberation. It could not filter out the interfering speakers. To this end, personalized speech enhancement (PSE) [1–4] is proposed to extract the voice of target speaker from all other speakers and background noise according to speech snippets enrolled from target speaker.

The latest ICASSP 2022 DNS challenge [5] aims to promote the full-band real-time personalized speech enhancement task. TEA-PSE [6] leads to superior performance on ICASSP 2022 DNS personalized speech enhancement evaluation set with a specifically designed two-stage framework. But it has a high computational complexity of 27.84 G multiply-accumulate operations (MACs) per second and performs on the full-band signal directly with a 0.96 real-time factor (RTF). Additionally, the encoder-decoder structure used in TEA-PSE could not capture long-range correlations effectively because of the constrained receptive field of convolutions according to [7].

To reduce the computation complexity, an intuitive way is feature compression. For example, RNNoise [8] and Personalized PercepNet [9] compress the full-band input feature using Bark-scale and equivalent rectangular bandwidth (ERB) scale, respectively. Such feature compression methods may inevitably lose crucial frequency band information, leading to sub-optimal performance. The second way is spectrum splitting, which is common in recent speech enhancement (SE) research. Lv *et al.* [10] and Li *et al.* [11] conduct spectrum splitting after the short-time Fourier transform (STFT), taking the stacked sub-bands as a batch instead of directly modeling the full-band feature. Different from such batch processing methods, DMF-Net [12] and SF-Net [13] use spectrum splitting with cascaded structure. When processing the higher band, the pre-processed lower band would give external knowledge guidance. The third way is finite impulse response (FIR) based sub-band analysis and synthesis, which can effectively reduce bandwidth in classic digital signal processing [14]. Multi-band WaveRNN [15] and multi-band MelGAN [16] have attained high MOS results using sub-band processing for text-to-speech (TTS) task. Such sub-band processing [17] used for the music source separation (MSS) task impressively outperforms the full-band processing by a significant margin.

On the other hand, the recent multi-stage approaches have shown superior performance with the intuitive assumption

†: Work done during an internship at Tencent Ethereal Audio Lab.
*: Corresponding author.

978-1-6654-7189-3/22/$31.00

that the original complicated speech enhancement problem can be decomposed into multiple simpler sub-problems and a better solution can be obtained in each stage progressively. Despite the specifically designed model architectures, we also notice that the optimization strategies adopted in these approaches are very different. Specifically, SDD-Net [18] and TEA-PSE [6] freeze the modules from the previous stage while training the current module. Differently, CTS-Net [19] finetunes the previous modules with a lower learning rate. Besides the staged training, Wang *et al.* [20] adopt an end-to-end training method, where different modules are simultaneously optimized with a single loss function. A comparative study is stilled desired to find the best training strategy for the multi-stage approaches.

In this paper, we propose TEA-PSE 2.0, to further improve perceived speech quality while significantly suppressing noise and interference and reducing computational complexity. Our contribution is threefold. First, we expand the original TEA-PSE model with sub-band processing using designed FIR filters and direct spectrum splitting. The sub-band input achieves better speech enhancement performance than full-band input as well as speeds up model inference significantly. Second, we upgrade the model with the time frequency convolution module (TFCM) [21] to increase the model's receptive field with small convolution kernels. Experiments show that our improved two-stage network results in substantial performance gain. Finally, we compare sequential optimization with joint optimization for multi-stage approaches. Single-domain and multi-domain loss functions are also investigated for PSE. Results show the superiority of sequential optimization training strategy and multi-domain loss function. Finally, the proposed TEA-PSE 2.0 outperforms the previous TEA-PSE with a 0.102 OVRL personalized DNS-MOS (PDNSMOS) [22] score improvement on the ICASSP 2022 DNS-challenge blind test set. Impressively, it only has 21.9% MACs of TEA-PSE in computation complexity.

2. PROBLEM FORMULATION

Let y be a mixture of target speaker, interference speaker and background noise captured by a single microphone in the time domain:

$$y(n) = s(n) * h_s(n) + z(n) * h_z(n) + v(n), \quad (1)$$

where n denotes the time sample index, s denotes the signal of the target speaker, z denotes the signal of the interference speaker, h_s and h_z denote the room impulse response (RIR) between the speaker and the microphone, and v denotes the additive noise. We use e to represent the target speaker's enrollment speech. In the frequency domain, Eq.(1) can be formulated as:

$$Y(t,f) = S(t,f) \cdot H_s(t,f) + Z(t,f) \cdot H_z(t,f) + V(t,f), \quad (2)$$

where t and f are the frame index and the frequency index, respectively. We don't explicitly consider dereverberation in our paper. During experiments, background noise, interference speech, and reverberation may exist at the same time.

3. TEA-PSE 2.0

3.1. Sub-band decomposition

In Fig. 1, we show the overall flow chart of TEA-PSE 2.0. We design two flows to compare the modeling abilities of different sub-band processing strategies. The first flow is the red line, which is based on FIR filter banks to do sub-band analysis and synthesis at signal level. The second flow is the green line, which is spectrum splitting and merging. In detail, K means the sampling interval, $F_M \in \mathbb{R}^{M\times M}$ means STFT kernel, $F_M^{-1} \in \mathbb{R}^{M\times M}$ indicates iSTFT kernel, and M is window length.

FIR filter banks analysis and synthesis (FAS). We adopt a stable and efficient design of filter bank – pseudo quadrature mirror filter bank (PQMF) [14], for sub-band decomposition and signal reconstruction. Both analysis and synthesis include K FIR filter banks, where K stands for the number of sub-band. Sub-band encoder has 3 procedures, including FIR analysis, down-sampling, and STFT. We use y_k to denote the output of the analysis and down-sampling, where $k \in [1, K]$ is the sub-band index. The sample rate of y_k is $\frac{1}{K}$ of y. $Y_k \in \mathbb{C}^{T\times F'}$ denotes the frequency domain corresponding to y_k. Then we stack Y_k along the channel dimension to form the feature $Y_{fas} \in \mathbb{C}^{T\times F'\times K}$ as the input of the PSE module. Sub-band decoder has 3 procedures, including iSTFT, up-sampling, and FIR synthesis. We slice the output of the PSE module $\hat{S}_{fas} \in \mathbb{C}^{T\times F'\times K}$ along the channel dimension as the prediction of each sub-band output $\hat{S}_k \in \mathbb{C}^{T\times F'}$. After iSTFT and up-sampling, $\hat{S}_k$ becomes $\hat{s}_k$, and the sample rate of $\hat{s}_k$ is the same as y. Finally, we pass $\hat{s}_k$ through a set of synthesis filter banks to reconstruct the source signal $\hat{s}$.

Spectrum splitting and merging (SSM). We split the full-band spectrum $Y \in \mathbb{C}^{T\times F}$ along the frequency axis for each sub-band $Y_k \in \mathbb{C}^{T\times \frac{F}{K}}, k \in [1, \ldots, K]$ and then stack them along the channel axis to form the input feature $Y_{ssm} \in \mathbb{C}^{T\times \frac{F}{K}\times K}$ as the input of the PSE module. During signal reconstruction, we reshape the network outputs $\hat{S}_{ssm} \in \mathbb{C}^{T\times \frac{F}{K}\times K}$ to recover full-band spectrum $\hat{S} \in \mathbb{C}^{T\times F}$.

3.2. Two-stage PSE network

The detail of the PSE module is shown in Fig. 2(a). We keep the same two-stage framework as TEA-PSE [6], which includes MAG-Net and COM-Net to process magnitude and complex features respectively. Fig. 2(b) shows the detail of the MAG-Net of the PSE module. We use E to represent speaker embedding extracted by the speaker encoder network. For MAG-Net, we use the magnitude of the observed signal Y as the input and the target magnitude as the training

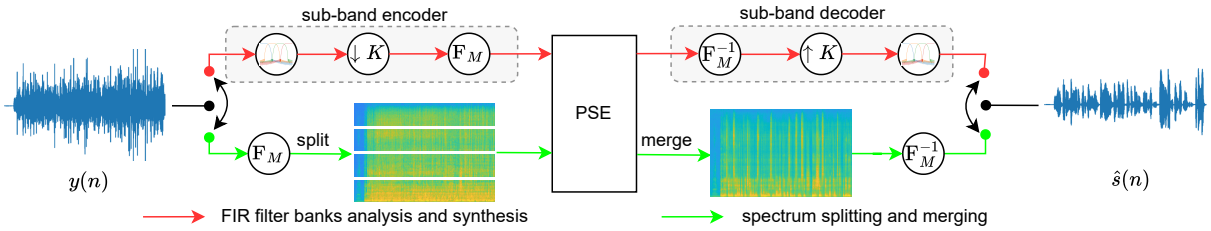

Fig. 1. Overall flow chart of TEA-PSE 2.0.

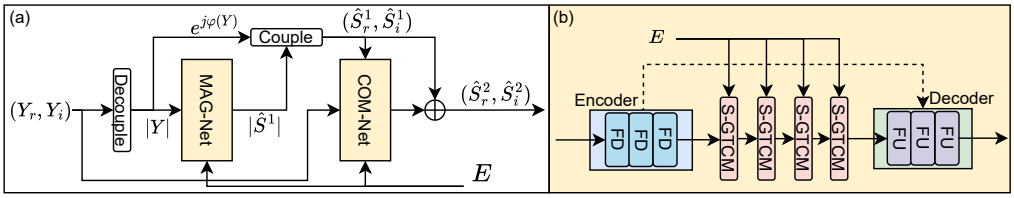

Fig. 2. (a) Detail of the PSE module. (b) The network detail of MAG-Net.

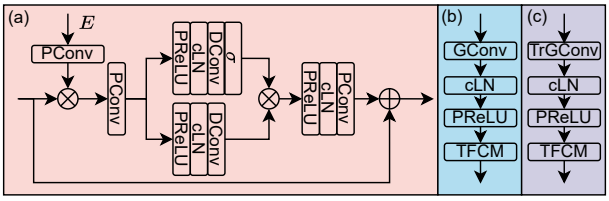

Fig. 3. (a) Detail of GTCM. (b) Detail of the FD layer in the encoder. (c) Detail of the FU layer in the decoder.

target. MAG-Net suppresses noise components and interference speech coarsely. Then, we couple $\hat{S}_1$ together with the noisy phase $e^{j\varphi(Y)}$ to get the real and imaginary (RI) spectrum $(\hat{S}_r^1, \hat{S}_i^1)$ as the input of COM-Net. We also adopt the observed spectrum (Y_r, Y_i) as the input of COM-Net to further remove the residual noise and interfering speech components. Residual connection is applied between the input $(\hat{S}_r^1, \hat{S}_i^1)$ and output of COM-Net to form the final output $(\hat{S}_r^2, \hat{S}_i^2)$. COM-Net has a similar network topology as MAG-Net, while its dual-decoder architecture is designed to estimate the RI spectrum separately.

Encoder and decoder. The encoder consists of three frequency down-sampling (FD) layers. Fig. 3(b) shows the detail of the FD layer. Gated Conv (GConv) [23, 24] is followed by the cumulative layer norm (cLN) [25], PReLU and time frequency convolution module (TFCM) [21]. We use TFCM to increase the receptive field with small convolution kernels, to solve the limited receptive field problem in the original convolutional encoder-decoder structure [7]. The decoder has three frequency up-sampling (FU) layers, whose architecture is shown in Fig. 3(c) in detail. It uses a mirror structure as FD layer and transposed gated Conv (TrGConv) to replace GConv.

Stacked gated temporal convolutional module. Our stacked gated temporal convolutional module (S-GTCM) repeats a stack of GTCM layers, as shown in Fig. 3(a), which maintains the same architecture as TEA-PSE [6]. GTCM contains two pointwise convolutions (PConv) and a dilated convolution (DConv). Between adjacent convolutions, PReLU and cLN are interpolated. Residual connection is applied between the input and output for training deeper networks. In each S-GTCM layer, the first GTCM layer accepts the learned representations $X \in \mathbb{R}^{T \times F''}$ over the mixture speech as well as the speaker embedding $E \in \mathbb{R}^D$, while other GTCM layers only accept mixture speech features as input. The speaker embedding is firstly repeated along the time dimension to form $E' \in \mathbb{R}^{T \times D}$ and then passes a PConv to keep the same dimension as learned representations. The learned representation is then multiplied with the speaker embedding in the feature dimension.

3.3. Loss function

We first apply scale-invariant signal-to-noise ratio (SI-SNR) [25] loss:

$$\mathcal{L}_{\text{si-snr}} = 20\log_{10}\frac{\left\|\left(\hat{s}^T s / s^T s\right)\cdot s\right\|}{\left\|\left(\hat{s}^T s / s^T s\right)\cdot s - \hat{s}\right\|}, \tag{3}$$

where $\hat{s}$ is the estimated speaker.

The power-law compressed phase-aware (PLCPA) loss is then used. PLCPA is advantageous for both ASR accuracy and perceptual quality [26, 27]. It is mostly made up of the following two components: magnitude loss $\mathcal{L}_{\text{mag}}$ and phase loss $\mathcal{L}_{\text{pha}}$.

$$\mathcal{L}_{\text{mag}} = \frac{1}{T}\sum_{t}^{T}\sum_{f}^{F}||S(t,f)|^p - |\hat{S}(t,f)|^p|^2. \tag{4}$$

$$\mathcal{L}_{\text{pha}} = \frac{1}{T}\sum_{t}^{T}\sum_{f}^{F}||S(t,f)|^p e^{j\varphi(S(t,f))} - |\hat{S}(t,f)|^p e^{j\varphi(\hat{S}(t,f))}|^2. \tag{5}$$

We employ the asymmetric loss to penalize the estimated spectrum of the target speaker, which is beneficial in alleviating over suppression [28].

$$\mathcal{L}_{\text{asym}} = \frac{1}{T}\sum_{t}^{T}\sum_{f}^{F}|\text{ReLU}(|S(t,f)|^p - |\hat{S}(t,f)|^p)|^2. \quad (6)$$

The two-stage network is trained using the following losses. We only train the MAG-Net with $\mathcal{L}_1$ initially.

$$\mathcal{L}_1 = \mathcal{L}_{\text{si-snr}} + \mathcal{L}_{\text{mag}} + \mathcal{L}_{\text{asym}}, \quad (7)$$

Following that, the pre-trained parameters of MAG-Net are frozen to only optimize the COM-Net by

$$\mathcal{L}_2 = \mathcal{L}_{\text{si-snr}} + \mathcal{L}_{\text{mag}} + \mathcal{L}_{\text{pha}} + \mathcal{L}_{\text{asym}}. \quad (8)$$

$\hat{S}$ and S are the estimated and clean spectrum, respectively. The spectral compression factor p is set to 0.5. Operator φ calculates the phase of a complex number.

4. EXPERIMENTS

4.1. Datasets

We use the ICASSP 2022 DNS-challenge full-band dataset [5] for experimentation. Approximately 750 hours of clean speech and 181 hours of noise clips make up the personalized DNS (PDNS) training set. There are 3230 speakers in total. The noise data originates from the DEMAND [31], Freesound, and Audio Set [32] databases.

We employ 700 hours of clean speech data and 150 hours of noise data, both taken from the PDNS dataset to generate the training set. 50 hours of clean speech data and 15 hours of noise data are selected to generate the development set. We generate 100,000 room impulse responses (RIRs) based on the image method [33] with RT60 $\in [0.1, 0.6]$s. The sizes of the room can be represented as $w \times d \times h$, where $w \in [3, 8]$m, $d \in [3, 8]$m, and $h \in [3, 4]$m. The microphone is dispersed across the space with h_{mic} ranging from $[0, 1.2]$m. Speech sound sources can be found anywhere in the room, with h_{speech} ranging from $[0.6, 1.8]$m. The distance between the sound source and microphone ranges from $[0.3, 6.0]$m. There are 80,000 and 10,000 RIRs in the training set and development set, respectively.

The evaluation data can be divided into two parts. The *simulation set* intends to evaluate the model's performance on unseen speakers. We use the KING-ASR-215 dataset as the source speech, the remaining 16 hours of data from the PDNS noise set as source noise, and the remaining 10,000 RIRs to create a simulation set of 2,010 noisy-clean pairings with 201 speakers. A random interfering speaker with SIR range of $[-5, 20]$dB and a random type of noise with SNR range of $[-5, 20]$dB are added to each noisy-clean pair. The second part is the *DNS2022 blind test set* which consists of 859 real test clips, in which 121 clips have interference speech and the remaining 738 clips have no interference speech. It should be noted that there is no overlap in the source data between the training, development, and simulated evaluation sets.

4.2. Training setup

The training data are generated on the fly and segmented into 10 s chunks in each batch, with SNR and SIR ranges of $[-5, 20]$ dB and $[-5, 20]$ dB, respectively. The scale for mixtures is adjusted to $[-35, -15]$ dBFS. Furthermore, 50% of the source speech data during training is convolved with RIR to simulate the reverb scenario. Additionally, 20% of the training data contain only one interfering speaker, 30% contain one interfering speaker and one type of noise, 30% contain only one type of noise, and the remaining 20% contain two types of noise.

We employ a Hanning window on the observed signal with a 20 ms frame length and 10 ms frame shift. The STFT contains 1024 points for the observed signal input with 48 kHz sampling rate, resulting in 513-dim features. The Adam optimizer [34] is used to optimize each neural model, using a $1e^{-3}$ initial learning rate. The learning rate is halved if the validation loss has no decrease for two epochs. Every stage is trained with 60 epochs. We apply a maximum l2 norm of 5 for gradient clipping.

The encoder has 3 FD layers and the decoder has 3 FU layers. The kernel size and stride of GConv and TrGConv are set as (1, 7) and (1, 4) in the time and frequency axis, respectively. The stride size along the frequency axis is 3 and 2 for experiments with 4 sub-bands and 8 sub-bands, respectively. For all GConv and TrGConv, the number of channels remains at 80. One TFCM contains 6 convolutional layers with a dilation rate of $\{$1,2,4,8,16,32$\}$. The kernel size of depthwise DConv in each TFCM is (3, 3), and the number of channels for all Conv in each TFCM is 80. The number of channels for all sub-modules in GTCM is set to 80 except for the last PConv. One S-GTCM contains 4 corresponding GTCM layers with a kernel size of 5 for DConv and a dilation rate of $\{$1,2,5,9$\}$, respectively. We stack 4 S-GTCM blocks to establish long-term relationships between consecutive frames as well as combine speaker embedding.

To extract speaker embedding from the enrollment speech of the target speaker, we use the pre-trained open-source ECAPA-TDNN [35] network.

4.3. Evaluation metrics

Several objective metrics are used, including wide-band perceptual evaluation speech quality (PESQ) [36] for speech quality, short-time objective intelligibility (STOI) [37], and its extended version ESTOI [38] for intelligibility, and SISNR [39] for speech distortion. We utilize the non-intrusive subjective evaluation metrics to evaluate the subjective speech performance, namely PDNSMOS P.835 [22], which are based on ITU-T P.835 [40]. The PDNSMOS P.835 is specifically designed for the PSE task, more details could be found in [22]. For all the metrics, high values indicate better performance.

Table 1. Performance on the simulation set in terms of PESQ, STOI (%), ESTOI (%), and SISNR (dB).

Id	Method	Params(M)	MACs(G/s)	Interference&Noise				Interference				Noise			
-	-	-	-	PESQ	STOI	ESTOI	SISNR	PESQ	STOI	ESTOI	SISNR	PESQ	STOI	ESTOI	SISNR
F1	Noisy	-	-	1.395	69.90	55.36	1.559	2.061	82.36	73.64	7.996	1.791	82.14	70.90	7.478
F2	TEA-PSE [6]	7.81	27.84	2.083	80.64	68.54	8.244	2.764	89.02	82.43	12.286	2.803	89.80	81.87	14.544
F3	MAG&COM-Net	8.59	16.93	**2.157**	**81.43**	**69.42**	**8.479**	**2.880**	**89.63**	**83.14**	**12.491**	**2.880**	**90.45**	**82.73**	**14.808**
F4	F3(-TFCM)	7.57	10.22	1.970	79.14	66.31	7.635	2.601	87.76	80.53	11.551	2.654	88.59	79.95	13.831
F5	MAG-Net	4.03	6.31	2.054	81.37	68.49	8.015	2.742	88.95	82.84	12.169	2.774	90.37	82.16	14.393
F6	COM-Net	4.56	10.6	2.116	81.38	69.17	8.283	2.825	89.19	82.96	12.279	2.841	90.37	82.41	14.611
S1	F3-SSM(K=2)	6.3	8.63	2.148	81.61	69.39	8.604	2.859	90.21	83.52	12.752	2.860	90.28	82.41	14.791
S2	F3-SSM(K=4)	5.75	6.11	**2.191**	**82.66**	**70.84**	9.040	**2.903**	**90.95**	**84.66**	13.144	**2.870**	**90.52**	**82.92**	14.902
S3	F3-SSM(K=8)	6.37	5.09	2.186	82.55	70.52	**9.248**	2.897	90.57	84.15	**13.307**	2.857	90.43	82.64	**15.014**
A1	F3-FAS(K=2)	6.3	8.63	**2.205**	82.14	70.26	8.795	**2.902**	90.37	83.99	12.888	**2.886**	90.39	**82.72**	14.839
A2	F3-FAS(K=4)	5.75	6.11	2.171	**82.27**	**70.32**	8.874	2.886	**90.64**	**84.30**	13.055	2.854	**90.44**	**82.72**	**14.881**
A3	F3-FAS(K=8)	6.37	5.2	2.140	81.56	69.15	**8.887**	2.846	90.29	83.52	**13.188**	2.816	89.92	81.80	14.837
M1	A2-freeze	5.75	6.11	2.171	82.27	70.32	8.874	2.886	**90.64**	**84.30**	13.055	2.854	90.44	82.72	**14.881**
M2	A2-finetune	5.75	6.11	**2.186**	82.31	70.27	**8.931**	**2.909**	90.58	84.22	**13.060**	2.873	90.44	82.69	14.867
M3	A2-joint	5.75	6.11	2.166	81.93	69.78	8.665	2.876	90.08	83.59	12.688	2.855	90.38	82.52	14.673
M4	A2-only	5.75	6.11	2.182	**82.36**	**70.49**	8.812	**2.909**	90.53	84.16	12.943	**2.876**	**90.55**	**82.86**	14.853
L1	A2-tfloss	5.75	6.11	2.171	82.27	70.32	**8.874**	2.886	**90.64**	84.30	**13.055**	2.854	90.44	82.72	**14.881**
L2	A2-tloss	5.75	6.11	1.986	80.99	68.61	8.691	2.665	89.41	82.38	12.643	2.615	89.73	81.56	14.719
L3	A2-floss	5.75	6.11	**2.196**	**82.37**	**70.73**	8.811	**2.906**	90.63	**84.38**	12.870	**2.876**	**90.57**	**82.97**	14.715

5. RESULTS AND ANALYSIS

We design several sets of experiments to verify different aspects of performance, including a) full-band input (F1-F6), b) sub-band processing based on SSM (S1-S3), c) sub-band processing based on FAS (A1-A3), d) training strategy (M1-M4), and e) loss function (L1-L3). The MAG-Net (F5) is optimized with $\mathcal{L}_1$ and the COM-Net (F6) is optimized with $\mathcal{L}_2$. We assess the outcomes for NSNet2 [30] (F7) and DeepFilterNet2 [29] (F8) using open-source pre-trained models for the blind test set. Bold results indicate the best in each column and the overall best results are underlined.

5.1. Full-band ablation

From Table 1, the two-stage MAG&COM-Net (F3) outperforms other methods in all metrics. Comparing F3 with F2 (TEA-PSE), MAG&COM-Net outperforms TEA-PSE and it has only 60% MACs of TEA-PSE. Comparing F3 with F4 (MAG&COM-Net without TFCM), the TFCM module can effectively expand the receptive field to learn the temporal correlations in both encoder and decoder. In Table 2, TFCM brings 0.116 SIG, 0.081 BAK, and 0.136 OVRL performance improvement. Comparing F5 (MAG-Net) and F6 (COM-Net) with F3, we use the two-stage network to further suppress noise components and unwanted interfering speaker voices, which is proven effective for the PSE task. For the DNS blind test set in Table 2, the MAG&COM-Net outperforms other baseline systems.

5.2. Sub-band ablation

FAS- and SSM-based sub-band processing outperform full-band features while greatly reducing computational complexity. This is because the sub-band input enables the model to assign different capabilities to different sub-bands [17], while models with full-band input tend to have high-frequency losses due to differences in high- and low-frequency energy [41]. For SSM-based sub-band processing, the performance of S2 (SSM, K=4) and S3 (SSM, K=8) are similar and are better than that of S1 (SSM, K=2). This can be attributed to the model's potential for better frequency modeling ability as the stride gets smaller. For FAS-based sub-band processing, the signal reconstruction error will become larger as the number of sub-band increases. So in Table 1, A3 (FAS, K=8) is less effective than A1 (FAS, K=2) and A2 (FAS, K=4). In general, 4 sub-band input performs best among the sub-band input experiments.

5.3. Training strategy ablation

A2 serves as the backbone for our subsequent experiments in ablation studies of training strategies and loss functions. We explore 4 different two-stage network training strategies, including M1): pre-train MAG-Net and then freeze it when training COM-Net, M2): pre-train MAG-Net and then fine-tune it with a smaller learning rate when training COM-Net, M3): train MAG-Net and COM-Net simultaneously using $\mathcal{L}_1 + \mathcal{L}_2$, and M4): train the whole network only using $\mathcal{L}_2$. In general, M1 (freeze) provides the best overall performance, as shown in Table 2, and M2 (finetune) also yields outstanding performance. For M4 (only), performance is negatively

Table 2. Performance on the DNS2022 blind test set. PDNS-MOS P.835 metrics include speech quality (SIG), background noise quality (BAK), and overall quality (OVRL).

Id	Method	Interference			Without Interference		
-	-	SIG	BAK	OVRL	SIG	BAK	OVRL
F1	Noisy	**3.973**	1.821	2.228	**4.224**	2.297	2.740
F2	TEA-PSE [6]	3.693	3.636	3.163	4.161	4.436	3.891
F3	MAG&COM-Net	3.714	**3.796**	**3.254**	4.173	4.452	**3.912**
F4	F3(-TFCM)	3.595	3.535	3.028	4.056	4.407	3.788
F5	MAG-Net	3.649	3.450	3.036	4.086	4.288	3.764
F6	COM-Net	3.537	3.542	3.003	4.107	4.408	3.834
F7	DeepFilterNet2 [29]	3.560	2.938	2.677	4.089	**4.487**	3.851
F8	NSNet2 [30]	3.467	2.713	2.435	3.725	4.179	3.415
S1	F3-SSM(K=2)	3.646	3.801	3.196	4.135	4.461	3.881
S2	F3-SSM(K=4)	3.859	**3.900**	3.389	4.206	**4.481**	**3.952**
S3	F3-SSM(K=8)	**3.896**	3.864	**3.393**	**4.209**	4.463	3.945
A1	F3-FAS(K=2)	3.853	3.868	3.371	4.212	**4.503**	3.966
A2	F3-FAS(K=4)	**3.859**	3.828	3.359	**4.229**	4.497	**3.979**
A3	F3-FAS(K=8)	3.845	**3.941**	**3.396**	4.186	4.482	3.933
M1	A2-freeze	**3.859**	3.828	**3.359**	**4.229**	**4.497**	**3.979**
M2	A2-finetune	3.801	3.862	3.338	4.199	4.474	3.941
M3	A2-joint	3.802	**3.883**	3.342	4.157	4.462	3.900
M4	A2-only	3.716	3.743	3.224	4.175	4.462	3.915
L1	A2-tfloss	3.859	**3.828**	**3.359**	**4.229**	**4.497**	**3.979**
L2	A2-tloss	**3.874**	3.712	3.293	4.177	4.394	3.860
L3	A2-floss	3.809	3.640	3.224	4.218	4.482	3.957

impacted by only optimizing $\mathcal{L}_2$, indicating the importance of placing optimization limits on all modules. For M3 (joint), the joint training strategy has the overall worst performance in Table 1. Based on the experimental results above, we conclude that sequential training, which means optimizing network modules one by one, is a preferred strategy for two-stage models in the PSE task, even though it will take more training time.

5.4. Loss function ablation

The time domain loss function is often used in the speech separation task, while the frequency domain loss function is commonly used in the speech enhancement task. To find the optimal loss strategy in the PSE task, we conduct experiments with loss functions in different domains, including L1): combine time domain loss function $\mathcal{L}_{\text{si-snr}}$ with frequency domain loss function $\mathcal{L}_{\text{mag}}$, $\mathcal{L}_{\text{pha}}$, and $\mathcal{L}_{\text{asym}}$, L2): time domain loss function, and L3): frequency domain loss function. In Table 2, L1 (tfloss) performs the best overall. This demonstrates the value of optimizing the model from both the time and frequency domains. In Table 1, L3 (floss) has the highest PESQ, STOI, and ESTOI scores.

6. CONCLUSIONS

In this paper, we propose a novel sub-band two-stage personalized speech enhancement network – TEA-PSE 2.0, which is an upgrade of TEA-PSE. Benefiting from the FAS-based sub-band processing and the TFCM module, TEA-PSE 2.0 (A2 in Table 2), with only 21.9% MACs of TEA-PSE in computation complexity, achieves the state-of-the-art PDNS-MOS performance in the ICASSP 2022 PDNS blind test set. It is also proved that optimizing MAG-Net and COM-Net sequentially is the best training strategy. Furthermore, it is valuable to optimize the model with a loss function combined with both time domain and frequency domain. In the future, we will particularly consider speech dereverberation in PSE to further improve speech quality.

7. ACKNOWLEDGEMENT

The authors thank S. Lv and J. Sun at Northwestern Polytechnical University for helpful discussions on sub-band processing, and Y. Fu at Netease Inc. for useful discussions on MTFAA-NET implementation.

8. REFERENCES

[1] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. Weiss, Y. Jia, and I. L. Moreno, "VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking," in *Interspeech*, 2019, pp. 2728–2732.

[2] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černockỳ, "Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 4, pp. 800–814, 2019.

[3] C. Xu, W. Rao, E. S. Chng, and H. Li, "Spex: Multi-scale time domain speaker extraction network," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 1370–1384, 2020.

[4] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "SpEx+: A Complete Time Domain Speaker Extraction Network," in *Interspeech*, 2020, pp. 1406–1410.

[5] H. Dubey, V. Gopal, R. Cutler, A. Aazami, S. Matusevych, S. Braun, S. E. Eskimez, M. Thakker, T. Yoshioka, H. Gamper, et al., "ICASSP 2022 deep noise suppression challenge," in *ICASSP*. IEEE, 2022, pp. 9271–9275.

[6] Y. Ju, W. Rao, X. Yan, Y. Fu, S. Lv, L. Cheng, Y. Wang, L. Xie, and S. Shang, "TEA-PSE: Tencent-ethereal-audio-lab Personalized Speech Enhancement System for ICASSP 2022 DNS CHALLENGE," in *ICASSP*. IEEE, 2022, pp. 9291–9295.

[7] S. Zhao, B. Ma, K. N. Watcharasupat, and W. Gan, "FRCRN: Boosting feature representation using frequency recurrence for monaural speech enhancement," in *ICASSP*. IEEE, 2022, pp. 9281–9285.

[8] J. Valin, "A hybrid dsp/deep learning approach to real-time full-band speech enhancement," in *MMSP*. IEEE, 2018, pp. 1–5.

[9] R. Giri, S. Venkataramani, J. Valin, U. Isik, and A. Krishnaswamy, "Personalized PercepNet: Real-Time, Low-Complexity Target Voice Separation and Enhancement," in *Interspeech*, 2021, pp. 1124–1128.

[10] S. Lv, Y. Hu, S. Zhang, and L. Xie, "DCCRN+: Channel-Wise Subband DCCRN with SNR Estimation for Speech Enhancement," in *Interspeech*, 2021, pp. 2816–2820.

[11] J. Li, D. Luo, Y. Liu, Y. Zhu, Z. Li, G. Cui, W. Tang, and W. Chen, "Densely connected multi-stage model with channel wise subband feature for real-time speech enhancement," in *ICASSP*. IEEE, 2021, pp. 6638–6642.

[12] G. Yu, Y. Guan, W. Meng, C. Zheng, and H. Wang, "DMF-Net: A decoupling-style multi-band fusion model for real-time full-band speech enhancement," *arXiv preprint arXiv:2203.00472*, 2022.

[13] G. Yu, A. Li, W. Liu, C. Zheng, Y. Wang, and H. Wang, "Optimizing shoulder to shoulder: A coordinated sub-band fusion model for real-time full-band speech enhancement," *arXiv preprint arXiv:2203.16033*, 2022.

[14] T. Q. Nguyen, "Near-perfect-reconstruction pseudo-qmf banks," *IEEE Trans. Signal Process.*, vol. 42, no. 1, pp. 65–76, 1994.

[15] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei, D. Su, and D. Yu, "DurIAN: Duration Informed Attention Network for Speech Synthesis," in *Interspeech*, 2020, pp. 2027–2031.

[16] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen, and L. Xie, "Multi-band MelGAN: Faster waveform generation for high-quality text-to-speech," in *SLT*. IEEE, 2021, pp. 492–498.

[17] H. Liu, L. Xie, J. Wu, and G. Yang, "Channel-Wise Subband Input for Better Voice and Accompaniment Separation on High Resolution Music," in *Interspeech*, 2020, pp. 1241–1245.

[18] A. Li, W. Liu, X. Luo, G. Yu, C. Zheng, and X. Li, "A Simultaneous Denoising and Dereverberation Framework with Target Decoupling," in *Interspeech*, 2021, pp. 2801–2805.

[19] A. Li, W. Liu, C. Zheng, C. Fan, and X. Li, "Two heads are better than one: A two-stage complex spectral mapping approach for monaural speech enhancement," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 1829–1843, 2021.

[20] H. Wang and D. Wang, "Cross-domain speech enhancement with a neural cascade architecture," in *ICASSP*. IEEE, 2022, pp. 7862–7866.

[21] G. Zhang, L. Yu, C. Wang, and J. Wei, "Multi-scale temporal frequency convolutional network with axial attention for speech enhancement," in *ICASSP*. IEEE, 2022, pp. 9122–9126.

[22] C. K. Reddy, V. Gopal, and R. Cutler, "DNSMOS P. 835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *ICASSP*. IEEE, 2022, pp. 886–890.

[23] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*. PMLR, 2017, pp. 933–941.

[24] S. Zhang, Z. Wang, Y. Ju, Y. Fu, Y. Na, Q. Fu, and L. Xie, "Personalized Acoustic Echo Cancellation for Full-duplex Communications," *arXiv preprint arXiv:2205.15195*, 2022.

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

[26] S. E. Eskimez, X. Wang, M. Tang, H. Yang, Z.n Zhu, Z. Chen, H. Wang, and T. Yoshioka, "Human Listening and Live Captioning: Multi-Task Training for Speech Enhancement," in *Interspeech*, 2021, pp. 2686–2690.

[27] S. Zhang, Z. Wang, J. Sun, Y. Fu, B. Tian, Q. Fu, and L. Xie, "Multi-task deep residual echo suppression with echo-aware loss," in *ICASSP*. IEEE, 2022, pp. 9127–9131.

[28] Q. Wang, I. L. Moreno, M. Saglam, K. Wilson, A. Chiao, R. Liu, Y. He, W. Li, J. Pelecanos, M. Nika, and A. Gruenstein, "VoiceFilter-Lite: Streaming Targeted Voice Separation for On-Device Speech Recognition," in *Interspeech*, 2020, pp. 2677–2681.

[29] H. Schröter, T. Rosenkranz, A. Maier, et al., "Deepfilternet2: Towards real-time speech enhancement on embedded devices for full-band audio," *arXiv preprint arXiv:2205.05474*, 2022.

[30] S. Braun and I. Tashev, "Data augmentation and loss normalization for deep noise suppression," in *International Conference on Speech and Computer*. Springer, 2020, pp. 79–86.

[31] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings," in *Proceedings of Meetings on Acoustics ICA2013*. Acoustical Society of America, 2013, vol. 19, p. 035081.

[32] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*. IEEE, 2017, pp. 776–780.

[33] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *JASA*, vol. 65, no. 4, pp. 943–950, 1979.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Interspeech*, 2020, pp. 3830–3834.

[36] I. Union, "Wideband extension to recommendation p. 862 for the assessment of wideband telephone networks and speech codecs," *International Telecommunication Union, Recommendation P*, vol. 862, 2007.

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

[38] J. Jensen and C. H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 11, pp. 2009–2022, 2016.

[39] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr–half-baked or well done?," in *ICASSP*. IEEE, 2019, pp. 626–630.

[40] B. Naderi and R. Cutler, "Subjective Evaluation of Noise Suppression Algorithms in Crowdsourcing," in *Interspeech*, 2021, pp. 2132–2136.

[41] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *WASPAA*. IEEE, 2017, pp. 21–25.