Neural Networks 154 (2022) 13–21

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Compressing speaker extraction model with ultra-low precision quantization and knowledge distillation

Yating Huang [a,b,1], Yunzhe Hao [a,b,1], Jiaming Xu [a,c,*], Bo Xu [a,b,c,d]

[a] Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China
[b] School of Future Technology, University of Chinese Academy of Sciences, Beijing, China
[c] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[d] Center for Excellence in Brain Science and Intelligence Technology, CAS, Shanghai, China

A R T I C L E   I N F O

Article history:
Received 11 June 2021
Received in revised form 20 April 2022
Accepted 21 June 2022
Available online 27 June 2022

Keywords:
Speaker extraction
Quantization-aware training
Knowledge distillation
Parameter sharing

A B S T R A C T

Recently, our proposed speaker extraction model, WASE (learning When to Attend for Speaker Extraction) yielded superior performance over the prior state-of-the-art methods by explicitly modeling onset clue and regarding it as important guidance in speaker extraction tasks. However, it still remains challenging when it comes to the deployments on the resource-constrained devices, where the model must be tiny and fast to perform inference with minimal budget in CPU and memory while keeping the speaker extraction performance. In this work, we utilize model compression techniques to alleviate the problem and propose a lightweight speaker extraction model, TinyWASE, which aims to run on resource-constrained devices. Specifically, we mainly investigate the grouping effects of quantization-aware training and knowledge distillation techniques in the speaker extraction task and propose Distillation-aware Quantization. Experiments on WSJ0-2mix dataset show that our proposed model can achieve comparable performance as the full-precision model while reducing the model size using ultra-low bits (e.g. 3 bits), obtaining 8.97x compression ratio and 2.15 MB model size. We further show that TinyWASE can combine with other model compression techniques, such as parameter sharing, to achieve compression ratio as high as 23.81 with limited performance degradation. Our code is available at https://github.com/aispeech-lab/TinyWASE.

1. Introduction

With the emergence of deep learning technologies, a growing number of neural network-based methods have been proposed for the cocktail party problem (Borgström, Brandstein, Ciccarelli, Quatieri, & Smalt, 2021; Bronkhorst, 2000; Chen & Zhang, 2021; Cherry, 1953; Michelsanti et al., 2021). While researchers have shown that large neural networks can achieve superb performance, one important topic for network design, especially for the speaker extraction problem, is to compress model size so that such models can be properly deployed on low-resource platforms such as mobile and hearable devices.

Current methods could be divided into two sorts according to the formalized definition of tasks, speech separation and speaker extraction (Michelsanti et al., 2021; Wang & Chen, 2018). Suppose there are N sources, $\mathbf{x}_1(t)$, $\mathbf{x}_2(t)$, …, $\mathbf{x}_N(t)$, then the mixture $\mathbf{x}(t)$ can be formulated as:

$$\mathbf{x}(t) = \sum_{i=1}^{N} \mathbf{x}_i(t). \tag{1}$$

Speech separation aims to estimate N sources from the mixture, which causes two problems, uncertain number of source problem and label permutation problem (Yu, Kolbæk, Tan, & Jensen, 2017). Deep clustering (Hershey, Chen, Le Roux, & Watanabe, 2016) and permutation invariant learning (Yu et al., 2017) are proposed to solve these problems. For speaker extraction, it utilizes partial information from the target speaker as prior knowledge, which does not require knowing the exact number of speakers and avoids the label permutation problem inherently (Delcroix, Zmolikova, Kinoshita, Ogawa, & Nakatani, 2018). On some occasions, people prefer to focus on someone and attend to what is being said. It suggests that speaker extraction has great development and application prospects in practical scenarios. Consequently, there is a strong demand for model compression and lightweight architectures. According to the types of auxiliary information, speaker extraction models can be divided into three types, vision-assisted models (Afouras, Chung, & Zisserman, 2019; Chen et al., 2021; Chung, Choe, Chung, & Kang, 2020; Ephrat et al., 2018;

* Corresponding author at: Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China.
E-mail addresses: huangyating2016@ia.ac.cn (Y. Huang), haoyunzhe2017@ia.ac.cn (Y. Hao), jiaming.xu@ia.ac.cn (J. Xu).
1 Contributed equally.

https://doi.org/10.1016/j.neunet.2022.06.026
0893-6080/© 2022 Elsevier Ltd. All rights reserved.

Hu et al., 2021), azimuth-assisted models (Gu et al., 2019; Li, Xu, Mesgarani and Xu, 2021; Li et al., 2021) and voiceprint-assisted models (Delcroix et al., 2018; Wang et al., 2020; Xu, Rao, Chng, & Li, 2020). Vision-assisted models use audio and video synchronized cameras to catch the coupling and correlation between vision and speech, and use visual information, such as lip movements, facial expressions and etc., to estimate the corresponding acoustic signal. Azimuth-assisted models utilize multi-channel acoustic recordings to obtain the spatial information of the acoustic scenes and extract the speaker at the given direction. Both the vision-assisted models and the azimuth-assisted models require relatively high computation costs. On the other hand, voiceprint-assisted models only require a single microphone to extract the voice characteristics from the spectrum distribution of enrolled speech as prior knowledge. Therefore, compared with vision-assisted models and azimuth-assisted models, voiceprint-assisted models show priority in lower computation costs and more application scenarios.

We recently proposed a novel model that can learn **W**hen to **A**ttend for **S**peaker **E**xtraction, abbreviated as WASE (Hao, Xu, Zhang, & Xu, 2021). WASE introduces onset signal, which is the start time of the target speaker's voice activity, into the speaker extraction task. The onset signal is believed to be an important clue in the auditory scene analysis when humans are in complex auditory scenes (Shamma, Elhilali, & Micheyl, 2011; Szabó, Denham, & Winkler, 2016). More specifically, WASE takes full advantage of the given enrolled reference utterance and obtains the voiceprint of the target speaker, which further assists the onset detector to find the starting point of the target speaker's speech in the mixture. Experimental results showed that the model with voiceprint clue and onset clue can achieve a 0.85 SDR improvement compared with the baseline model only with voiceprint clue on WSJ0-2mix dataset (Hershey et al., 2016) with the same model size and the similar network structure. Moreover, WASE explores more complex clue forms, i.e., extending onset clue to onset and offset clues, and the experimental performances are further improved. Other current popular voiceprint-assisted models include Voicefilter (Wang et al., 2019), SpEx (Xu et al., 2020), SpeakerBeam (Žmolíková et al., 2019) and etc. Balancing the performance and model size, we finally choose WASE as the baseline model in our work.

It is worth noting that although WASE can perform well on both speaker extraction and speaker onset detection tasks, the model size and the memory consumption are still relatively high, which may pose constraints on its applications on resource-constrained platforms, such as mobile phones. Therefore, it is necessary to compress the models while preserving the performance. Various model compression methods have been investigated to compress deep neural networks, including quantization (Rastegari, Ordonez, Redmon, & Farhadi, 2016; Wu, Li, Chen, & Shi, 2018; Zhou et al., 2016), knowledge distillation (Ahn, Hu, Damianou, Lawrence, & Dai, 2019; Hinton, Vinyals, & Dean, 2015; Huang & Wang, 2017; Romero et al., 2014), pruning (Han, Mao, & Dally, 2016; He, Zhang, & Sun, 2017) and etc. However, existing deep model compression methods mainly focus on image classification or segmentation tasks, and less attention is paid to speaker extraction task, which is a regression task and needs to reconstruct speech with fine structures.

In this work, we propose TinyWASE, which combines quantization-aware training and knowledge distillation technologies to reduce the model size and complexity while maintaining the model performance compared to the baseline full-precision model. We investigate and conduct experiments on how to combine quantization and knowledge distillation to reduce the performance degradation to a minimum. These strategies may shed light on compressing models in the speech separation and speaker extraction tasks, and are not constrained to our previously proposed WASE model.

2. Related work

2.1. Model compression

In order to alleviate the deployment problem on resource-constrained devices, many deep model compression methods have been proposed to compress large deep learning models, including knowledge distillation (Hinton et al., 2015) and quantization (Rastegari et al., 2016; Wu et al., 2018; Zhou et al., 2016).

Knowledge distillation is first developed in Hinton et al. (2015) to transfer the knowledge from a large teacher model to a more compact small student model. The main idea is that the small student model mimics the behaviors of the teacher model under the supervision signal from the teacher model, which is referred to as "knowledge". The knowledge can be logits, activations or features from the hidden layers (Ahn et al., 2019; Hinton et al., 2015; Huang & Wang, 2017; Romero et al., 2014).

Different from knowledge distillation, quantization compresses a model by quantizing the weights and/or the activations to lower bits. Methods for quantizing the full-precision neural networks can be roughly divided into two types: approximation-based approaches (Rastegari et al., 2016; Wu et al., 2018; Zhou et al., 2016) and loss-aware-based approaches (Hou & Kwok, 2018; Leng, Dou, Li, Zhu, & Jin, 2018). Approximation-based approaches approximate the full-precision values with lower bits via step functions in the forward pass and additionally approximate the gradients in the backward pass, since the quantization operators are non-differentiable. As a result, the use of different approximations in the forward pass and the backward pass causes a gradient mismatch problem, which may make the training procedure of an ultra-low bit network unstable. In order to avoid the approximation of gradients, loss-aware-based methods incorporate the loss of the neural networks and directly formulate quantization as an optimization problem, which aims to minimize the training loss. However, loss-aware-based methods are only suitable for quantizing weights, but not for activations, and suffer from high computation costs during training. To alleviate the problems mentioned above, Yang et al. (2019) reformulate the quantization operation as a simple non-linear function, termed quantization function, so that quantization functions can be learned like activation functions in an end-to-end way. The work mentioned above mainly focuses on the image classification task and few researchers apply ultra-low bit quantization methods to the speech separation task, which is a regression task other than a classification task.

2.2. Model compression for speaker extraction

Speech separation/speaker extraction models based on deep learning have achieved impressive performance on the standard datasets, but they require high hardware storage space and computing power. Recently, some studies have focused on the compression techniques of the speech separation model for resource-constrained devices. Wang et al. (2020) propose a lightweight model, Voicefilter-lite. However, the aim of their work is to improve the performance of automatic speech recognition, which serves as the backend of the standard speech processing pipeline, and they only report the performance of automatic speech recognition, lacking the speech separation metrics. But there are some situations where users care more about the quality of the separated speech rather than the recognition results of the speech. For example, people wearing hearing aids may need the speech separation algorithm to help them separate the target speech from the noisy mixture. In addition, Voicefilter-lite only performs 8-bit post-training quantization and does not further explore the limit of the trade-off between the quantization bits and the performance. Tuan, Wu, Lee, and Tsao (2019) study the performance of parameter sharing strategies on TasNet. The results show that with parameter sharing strategy along the stack dimension, TasNet could achieve a compression ratio of 26.7% at the cost of less than 1 point of performance degradation on WSJ0-2mix dataset (Hershey et al., 2016). Chen, Liu, Shi, Xu, and Xu (2018) integrate knowledge distillation into the Binary Neural Network (BNN) training phase and achieve performance improvements. This is the first time that both of knowledge distillation and binarization have been used in speech separation. In addition to general compression techniques, some research focuses on the design of the architecture of lightweight models. Luo, Han, and Mesgarani (2021) propose Group Communication with Context Codec (GC3). The main idea of GC3 is to repeatedly use group communication modules to divide the high-dimensional features into low-dimensional groups, and use a small shared module to process low-dimensional features. For a sequence of features $\mathbf{H} \in \mathbf{R}^{N\times T}$, where N denotes the encoding dimension and T denotes the number of time steps, they split $\mathbf{H}$ in both the feature dimension and the temporal dimension. However, in order to get a comparable performance, the model architecture of GC3-based models needs to be carefully adjusted.

In this work, we use quantization functions to quantize the weights to ultra-low bits and use min–max linear quantization to quantize the activations to 8 bits, and combine knowledge distillation techniques to further improve the performance of speaker extraction. Similarly, Polino, Pascanu, and Alistarh (2018) investigate the grouping effects of quantization and knowledge distillation in image classification and machine translation. Our work is different from Polino et al. (2018) mainly in two aspects:

- We investigate the effects of the combination of ultra-low bit quantization-aware training and knowledge distillation in the speaker extraction task, which is a regression task rather than a classification task. To the best of our knowledge, this is the first time that the effects of ultra-low bit quantization and knowledge distillation have been studied in speaker extraction.
- We study the quantization scheme that not only quantizes the weights but also the activations while Polino et al. (2018) only focus on quantizing the weights. Since our experiments are based on a vanilla convolutional neural network (CNN) model, the combination and optimization of multiple compression techniques could provide reference and enlightenment for other CNN-based speaker extraction models.

3. WASE recap

In this section, we give a brief review of our previously proposed WASE (learning When to Attend for Speaker Extraction) model, which uses voiceprint and onset clues to extract the speech of the target speaker.

WASE contains five modules, namely encoder, decoder, voiceprint extractor, modulation module and speaker extraction module, as shown in Fig. 1. WASE adopts a common time-domain masking-based approach. Specifically, the mixture is encoded by the encoder to obtain the spectrum in the time domain. Then the speaker extraction modulator utilizes voiceprint and onset clues extracted from the reference utterance as the guide signal to generate the target speaker mask. Finally, the decoder decodes the masked spectrum to output the target speech.

We assume voiceprint, a vector describing the speaker's characteristics, is available at runtime without additional computation costs. In practice, there is usually an enrollment process before enabling a speaker extraction application. The voiceprint vector is computed from the target user's recordings and stored on the device in advance. In this paper, the voiceprint vector is computed from a voiceprint extractor, which is composed of a 1-dimensional (1-D) convolutional layer, a two-layer Bidirectional Long Short-Term Memory (BLSTM), a linear layer and a mean pooling layer in series. The voiceprint extractor processes the enrolled utterance of the target speaker and outputs the voiceprint.

For the main part of WASE network, as shown in Fig. 1, the encoder consists of a 1-D convolutional layer, and the decoder consists of a 1-D transpose convolutional layer. In the modulation module, the voiceprint first modulates the intermediate feature in the frequency domain, and then the onset clue vector modulates the intermediate feature in the time domain again. The modulated feature is further processed by the Temporal Convolutional Network (TCN) blocks in the speaker extraction module. The TCN block repeats R times and each repeat contains X 1-D convolutional blocks. The dilation factors of the convolutional blocks in each repeat increase exponentially by 2. Each 1-D convolutional block has two output paths, one path for the next 1-D convolutional block and the other for mask generation.

WASE takes a multi-task learning strategy to train the model. The main task of WASE is to separate the target signal from the mixture, which is supervised by Scale-Invariant Source-to-Noise Ratio (SI-SNR) between the predicted wave $\mathbf{x}_{pred}$ and the target clean wave $\mathbf{x}$. We define the construction loss as Eq. (2):

$$\begin{cases} \mathcal{L}_{construct} = -SI\text{-}SNR(\mathbf{x}_{pred}, \mathbf{x}), \\ SI\text{-}SNR(\mathbf{x}_{pred}, \mathbf{x}) = 10\log_{10}\frac{\|\mathbf{x}_{target}\|^2}{\|\mathbf{e}_{noise}\|^2}, \\ \mathbf{x}_{target} = \frac{\langle \mathbf{x}_{pred}, \mathbf{x}\rangle \mathbf{x}}{\|\mathbf{x}\|^2}, \\ \mathbf{e}_{noise} = \mathbf{x}_{pred} - \mathbf{x}_{target}. \end{cases} \tag{2}$$

Another task of WASE is onset detection, which uses the Cross-Entropy (CE) loss between the predicted onset vector $\mathbf{p}_{pred}$ and the target oracle onset vector $\mathbf{p}_{target}$.

$$\mathcal{L}_{ce} = CE(\mathbf{p}_{pred}, \mathbf{p}_{target}). \tag{3}$$

The loss function of WASE is defined as Eq. (4):

$$\mathcal{L} = \mathcal{L}_{construct} + \lambda\mathcal{L}_{ce}, \tag{4}$$

where λ balances the construction loss and the CE loss.

4. Model compression for speaker extraction

In this section, we explore model compression techniques for speaker extraction. We propose Distillation-aware Quantization in this section, and refer to the compressed WASE model as TinyWASE for simplicity. To further increase compression ratio, we explore to apply the parameter sharing strategy to the model.

4.1. Quantization

In this section, we introduce the quantization methods used in this paper.

4.1.1. Weight quantization

For weight quantization, we explore the quantization function (Yang et al., 2019) to quantize weights to ultra-low bits (e.g. 3 bits) so that we can train the low-bit neural network efficiently in an end-to-end way. The main idea of the quantization function is to reformulate the quantization operation using a linear combination of several simple functions and gradually approximate the quantization operation using a soft quantization function by controlling the temperature parameter during training. The low-bit quantization operation can be reformulated as a combination of

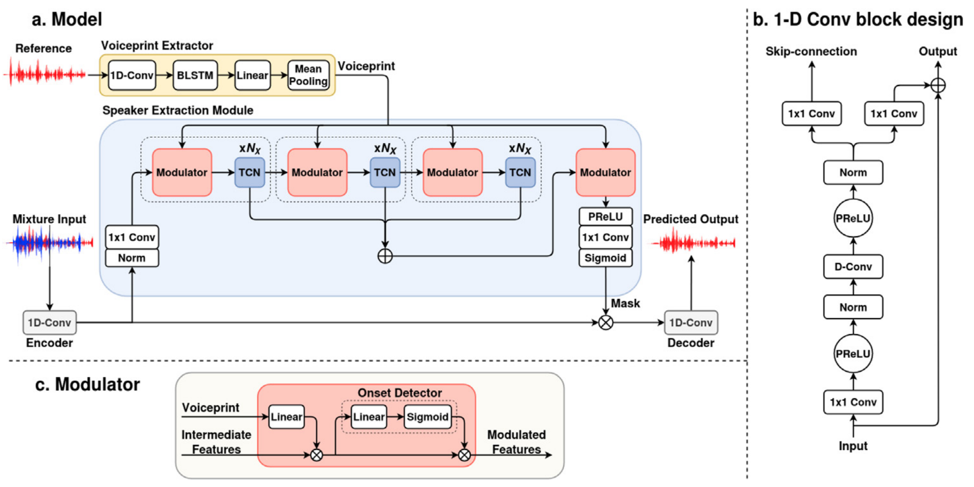


Fig. 1. An illustration of WASE model. *(a)* The overall architecture of WASE. *(b)* Details of 1-D convolutional blocks used in a TCN block. "D-Conv" indicates a depthwise convolution. *(c)* The structure of the modulator.

several unit step functions, termed as ideal quantization function. The ideal quantization function is non-differentiable due to the use of the unit step function. Suppose that x is the full-precision value to be quantized, y is the quantized integer constrained to a predefined set $\mathcal{Y} = \{\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_{n+1}\}$. For example, if we do 3-bit quantization, $\mathcal{Y}$ can be defined as $\mathcal{Y} = \{-3, -2, -1, 0, 1, 2, 3\}$. The ideal quantization function is formulated as Eq. (5):

$$y = \sum_{i=1}^{n} s_i \mathcal{A}(\beta x - b_i) - o, \tag{5}$$

$$\mathcal{A}(x) = \begin{cases} 1 & x \geq 0, \\ 0 & x < 0, \end{cases} \tag{6}$$

where $n = |\mathcal{Y}| - 1$, β is the scale factor of the input. Eq. (6) defines the unit step function, where $s_i = \mathcal{Y}_{i+1} - \mathcal{Y}_i$ and b_i is the bias for the unit step function. The global offset is defined as $o = \frac{1}{2}\sum_i^n s_i$.

By replacing the non-differentiable unit step function with sigmoid function, the ideal quantization function is transferred to a differentiable soft quantization function in the training procedure, which can be learned like an activation function in an end-to-end way, as shown in Eq. (7):

$$y = \mathcal{Q}(x) = \alpha(\sum_{i=1}^{n} s_i \sigma(T(\beta x_d - b_i)) - o), \tag{7}$$

$$\sigma(Tx) = \frac{1}{1 + \exp(-Tx)}, \tag{8}$$

where β and α are learnable scale factors of the input and the output, respectively. In the inference phase, the ideal quantization function Eq. (5) is used instead. The use of different quantization functions in the training phase and inference phase may cause performance degradation. To narrow the performance gap, temperature T is introduced to the sigmoid function. The gap between the soft quantization function and the ideal quantization is small when the temperature T is large and vice versa. If we use a large fixed T while training, the learning capacity of the model is constrained, as the gradients are zeros in most cases, which may fail to converge to a satisfactory solution. Thus, it is always a good practice to start from a small temperature T and gradually increase the temperature so that the quantized neural networks can be well learned and the gap between the training phase and the inference phase can be ignored at the end of the training procedure (Yang et al., 2019).

In this paper, we perform layer-wise non-uniform quantization. Weights from different layers use different quantization functions. Following the previous work (Yang et al., 2019), we do not quantize the first convolutional layer and the last transpose convolutional layer. We also do not quantize the PReLU function and the layer normalization function, as quantizing them would result in performance degradation and the number of parameters in these layers is negligible. Note that we assume that voiceprint vectors do not require additional computation costs and do not consider the quantization of the voiceprint extractor. In order to perform non-uniform quantization, we perform K-means clustering on the full-precision weights and get $n+1$ ranked centers of clusters, $c_1, c_2, \ldots, c_{n+1}$ in the ascending order. The bias b_i is initialized as $b_i = \frac{c_i + c_{i+1}}{2}$ and gets fixed during training, since it does not make much difference in learning biases in the quantization function.

4.1.2. *Activation quantization*

We also quantize activations to make the most expensive matrix multiplication faster. Different from the non-uniform weight quantization to ultra-low bits, we simply quantize the activations to 8 bits. We use the most commonly used 8-bit min–max linear quantization (Krishnamoorthi, 2018), which is defined as:

$$\mathcal{Q}(x) = round(\frac{x - \mathbf{x}_{min}}{s}), \tag{9}$$

$$s = \frac{\mathbf{x}_{max} - \mathbf{x}_{min}}{2^p - 1}, \tag{10}$$

where $\mathbf{x}$ is the tensor to be quantized, x is the element in $\mathbf{x}$ and $p = 8$ for 8-bit quantization. Straight-Through Estimator (STE) (Courbariaux, Bengio, & David, 2015) is used to back-propagate the gradients through quantized activations.

4.2. *Distillation-aware quantization*

Due to the relatively low capacity of low-bit quantized networks, there will be some performance degradation compared with the full-precision network of the same model structure. To further improve the performance of the quantized model, we incorporate the knowledge distillation technique in the training procedure. The framework of our proposed method is shown in Fig. 2. In our experiments, the full-precision WASE model acts as the teacher model and the quantized TinyWASE model acts as the student model so that the student attempts to learn the behaviors of the teacher model. To be specific, the distillation loss is defined

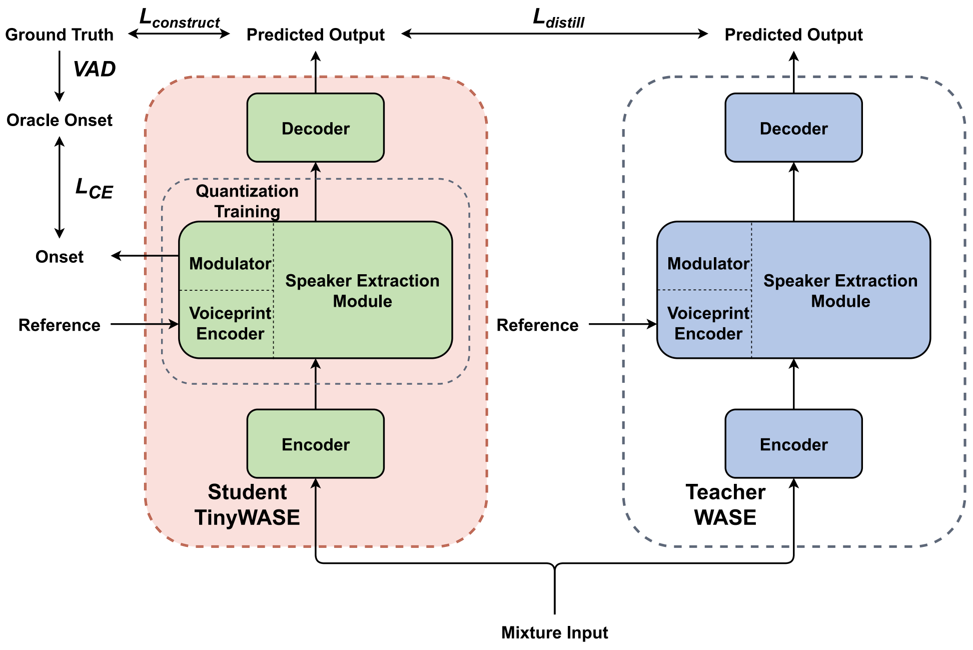


Fig. 2. The proposed procedure of quantization-aware training and knowledge distillation.

as the SI-SNR loss between the output of the teacher model $\mathbf{x}_{teacher}$ and the output of the student model $\mathbf{x}$, as shown in Eq. (11):

$$\mathcal{L}_{distill} = -SI\text{-}SNR(\mathbf{x}_{teacher}, \mathbf{x}). \quad (11)$$

Thus, the loss function of TinyWASE model totally consists of three parts, which is defined as:

$$\mathcal{L} = \mathcal{L}_{construct} + \lambda_1 \mathcal{L}_{CE} + \lambda_2 \mathcal{L}_{distill}, \quad (12)$$

where λ_1 and λ_2 are the hyperparameters balancing the losses. We use the full-precision WASE model to initialize our quantized student model. The whole training procedure is shown in Algorithm 1, termed as Distillation-aware Quantization.

4.3. Parameter sharing

To further increase the compression ratio, inspired by Tuan et al. (2019), we explore to apply parameter sharing strategy to TinyWASE. To be specific, the parameters are shared between R TCN blocks in the speaker extraction module and each TCN block contains X 1-D convolutional blocks. We first train a full-precision parameter-sharing model, then the weights of the pretrained model is loaded into the low-bit parameter-sharing model to initialize the weights. After that, Distillation-aware Quantization is performed.

5. Experiments

5.1. Experimental settings

We choose WASE with onset clue and voiceprint clue as our baseline model. The hyperparameters of voiceprint encoder and modulation modules are stated in Section 3, which are the same for all the models in our experiments. For convolutional encoder and decoder, a 50% stride size is used. Other hyperparameters of the network are shown in Table 1. The code for our experiments is publicly available at https://github.com/aispeech-lab/TinyWASE.

We perform experiments on speech separation and speaker extraction benchmark dataset WSJ0-2mix (Hershey et al., 2016). There are 101 speakers in the training set and 18 unseen speakers in the test set. The total length of training set, validation set and test set are 30 h, 10 h and 5 h, respectively. All samples are downsampled to 8 kHz.

Two speakers are randomly selected from the training set. We regard one as the target speaker and the other as the interference speaker. We generate the noisy training samples on the fly by mixing the clean utterance and the interference utterance at a random Signal-to-Noise Ratio (SNR) between −5 dB and 5 dB, and limit the length to 4 s. All the inputs are normalized using

Algorithm 1 Distillation-aware Quantization based on quantization functions

```
Input: A fixed teacher model; training dataset; network with
    M modules and their corresponding inputs/activations and
    trainable weights to be quantized.
Output: Quantized networks.
 1: Initialize the weights of the quantized student model using
    the full-precision pretrained model.
 2: for $m = 1, ..., M$ do
 3:   Initialize the biases of the quantization function.
 4: end for
 5: for $t = 1, ..., T_{max}$ do
 6:   Get next mini-batch of data.
 7:   for $m = 1, ..., M$ do
 8:     Apply soft quantization function Eq. (7) to the weights of
        module $m$.
 9:     Apply min-max linear quantization Eq. (9) to the
        inputs/activations of module $m$.
10:     Feedforward module m with the quantized weights and
        activations.
11:     Compute the loss function in Eq. (12).
12:     Backpropagate the gradients and train the parameters
        with gradually increasing temperature T.
13:   end for
14: end for
15: for $m = 1, ..., M$ do
16:   Replace the soft quantization function Eq. (7) with ideal
      quantization function Eq. (5) for inference.
17: end for
```

Table 1
Hyperparameters of TinyWASE. X and R will be specified in the different experimental settings in Section 5.2.

Symbol	Description	Value
N	Number of filters in mixture encoder/decoder	512
L	Length of the filters (in samples)	16
B	Number of channels in bottleneck and the residual paths' 1 × 1-conv blocks	128
S_C	Number of channels in skip-connection paths' 1 × 1-conv blocks	512
H	Number of channels in convolutional blocks	128
P	Kernel size in convolutional blocks	3
X	Number of convolutional blocks in each repeat	–
R	Number of repeats	–

the computed mean value and the standard deviation. The scale of the normalized inputs is [0, 1]. In the test phase, we set one of the two speakers in the original test set as the target speaker in turn and finally obtain 6000 samples, which is twice the number of the original test set.

Adam optimizer (Kingma & Ba, 2015) with an original learning rate of $1e^{-3}$ is used to train the full-precision WASE. To train TinyWASE, we first load the weights of the pretrained full-precision model to get a good starting point, then use Adam optimizer with a learning rate of $5e^{-4}$ to train TinyWASE for 50 epochs. Gradient clipping with maximum L_2-norm of 5 is applied during training. λ_1 and λ_2 in Eq. (12) are 1 and 0.2, respectively. The temperature T in Eq. (8) is set to 10 in the beginning and we increase T linearly with respect to the training epoch, which is $T = epoch \times 10$.

5.2. Effects of distillation-aware quantization

We quantize the weights of our networks to 3 bits and 4 bits, and quantize the activations of each layer to 8 bits. The teacher model is the full-precision WASE model with 3 TCN blocks and there are 8 1-D convolutional blocks in each TCN block, dubbed as 3 × 8 WASE model. The student model is the low-precision TinyWASE. We conduct experiments on student models with different model structures, namely the 3 × 8 TinyWASE model and the 3 × 4 TinyWASE model. We also explore to quantize the model without using knowledge distillation. The results are listed in Table 2. For both the 3 × 8 and the 3 × 4 TinyWASE model, models trained with knowledge distillation perform slightly better than those without knowledge distillation. Especially for the 3 × 4 TinyWASE model with 4-bit weights and 8-bit activations, the performance is almost the same as the full-precision 3 × 4 WASE model.

As shown in Table 3, for the space complexity of our model, we report the parameter numbers, the model size and the compression ratio of TinyWASE. Note that we do not consider the voiceprint extractor here, since we assume the voiceprint clue does not require extra computation costs as stated in Section 3. Combined with Table 2, we can conclude that TinyWASE achieves comparable results as full-precision models with a much smaller model size.

5.3. Effects of parameter sharing

To further compress the model size, we investigate the effects of parameter sharing. We quantize the weights to 3 bits and the activations to 8 bits. The parameters of the three TCN blocks are shared in this experiment. The results are listed in Table 4. We can see from the table that by combining parameter sharing, TinyWASE can achieve a larger compression ratio with some performance degradation. The performance degradation is largely due to parameter sharing but not Distillation-aware Quantization.

Table 2
Test set results of the proposed TinyWASE under different configurations on WSJ0-2mix dataset. "W" and "A" represent the quantization bits of the weights and the activations, respectively. -kd means that we train the model without using knowledge distillation.

Model (R, X)	W-A	SDR (dB)	SDRi (dB)
WASE (3, 8)	32-32	17.12	16.99
TinyWASE (3, 8)	4-8	16.65	16.52
TinyWASE (3, 8)-kd	4-8	16.60	16.47
TinyWASE (3, 8)	3-8	16.18	16.00
TinyWASE (3, 8)-kd	3-8	16.05	15.92
WASE (3, 4)	32-32	13.24	13.11
TinyWASE (3, 4)	4-8	13.23	13.09
TinyWASE (3, 4)-kd	4-8	13.11	12.98
TinyWASE (3, 4)	3-8	13.11	12.97
TinyWASE (3, 4)-kd	3-8	12.97	12.83

Table 3
Number of parameters and model size (× compression ratio) of TinyWASE.

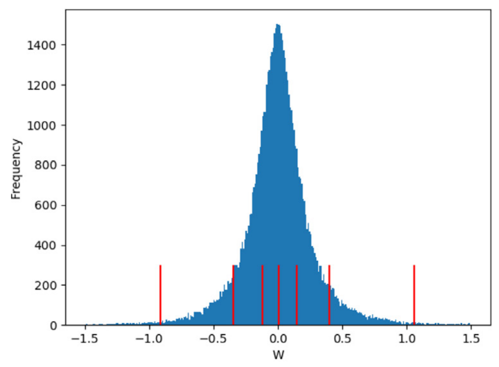
Model (R, X)	W-A	#Params	Size (×ratio)
TinyWASE (3, 8)	3-8	5.05M	2.15 (×8.97)
	4-8	5.05M	2.74 (×7.04)
TinyWASE (3, 4)	3-8	2.63M	1.15 (×8.77)
	4-8	2.63M	1.45 (×6.92)

Table 4
Effects of weight sharing. "W" and "A" represent the quantization bits of the weights and activations, respectively. The number of parameters, the model size (× compression ratio), SDR and SDRi are listed in the table. WS means weight sharing, SDR and SDRi are measured with dB.

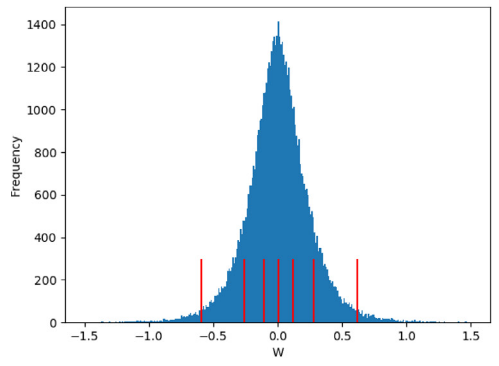
Model (R, X)	WS	W-A	#Params	Size (×ratio)	SDR	SDRi
WASE (3,8)	No	32-32	5.05M	19.29	17.12	16.99
TinyWASE (3,8)	No	3-8	5.05M	2.15 (×8.97)	16.18	16.00
WASE (3,8)	Yes	32-32	1.83M	6.97 (×2.77)	14.81	14.68
TinyWASE (3,8)	Yes	3-8	1.83M	0.81 (×23.81)	14.29	14.16
WASE (3,4)	No	32-32	2.63M	10.04	13.24	13.11
TinyWASE (3,4)	No	3-8	2.63M	1.15 (×8.77)	13.23	13.09
WASE (3,4)	Yes	32-32	1.02M	3.90 (×2.58)	12.66	12.52
TinyWASE (3,4)	Yes	3-8	1.02M	0.48 (×20.93)	12.04	11.92

5.4. Comparison with post-training quantization

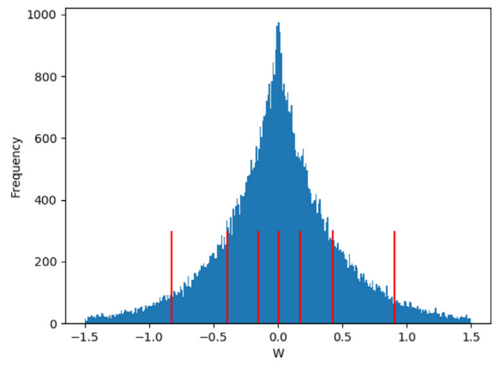
We compare our proposed method with the most commonly used linear quantization method (Krishnamoorthi, 2018), which is a post-training quantization method, on WSJ0-2mix. The results are listed in Table 5. We can see from Table 5 that Distillation-aware Quantization performs much better than the baseline over all experimental configurations, especially under low bits. There is a huge performance gap for the baseline post-training linear quantization method between 4 bits and 3 bits, whereas there is only a performance drop of about 0.5 between 4 bits and 3 bits for Distillation-aware Quantization.



(a) TCN1-1



(b) TCN2-1



(c) TCN3-1

Fig. 3. The weight distributions of the convolutional layers of the full-precision 3 × 8 WASE model. (a), (b) and (c) are the distributions of the first convolutional layers from the first TCN block, the second TCN block and the third TCN block before quantization, respectively. The red lines denote the $n + 1$ clustering centroids of K-Means clustering when the network is quantized to 3 bits. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.5. Results on GRID-mix

We also evaluate our system on another 2-mixture speech separation dataset GRID-mix dataset generated by GRID (Cooke, Barker, Cunningham, & Shao, 2006). GRID contains 3-second videos of 18 male speakers and 15 female speakers. We randomly select 3 males and 3 females to construct a validation set of 2.5 h and another 3 males and 3 females for a test set of 2.5 h. The rest of the speakers form the training set of 30 h. To construct a 2-speaker mixture, we randomly choose two different speakers first, randomly select audio from each chosen speaker, and finally mix two audio clips at a random SNR level between −5 dB and 5 dB.

We conduct experiments on 3 × 8 TinyWASE and 3 × 4 TinyWASE. The teacher model is the full-precision WASE model with the same model structure as the student model. The results are listed in Table 6. We can see from Table 6 that TinyWASE has the ability to achieve performance on par with the full-precision model on GRID-mix.

Table 5
Comparison of our proposed Distillation-aware Quantization and post-training linear quantization on WSJ0-2mix. "W" and "A" represent the quantization bits of the weights and the activations, respectively. The model has 3 TCN blocks and there are 8 convolutional blocks in each TCN block.

Quantization methods	W-A	SDR (dB)	SDRi (dB)
Distillation-aware Quantization	3-8	16.18	16.00
	4-8	16.65	16.52
	6-8	16.82	16.68
	8-8	16.93	16.80
Post-training linear quantization	3-8	4.88	4.75
	4-8	13.79	13.66
	6-8	15.67	15.53
	8-8	15.86	15.72

Table 6
Test set results of the proposed TinyWASE under different configurations on GRID-mix dataset. "W" and "A" represent the quantization bits of the weights and the activations, respectively.

Model (R, X)	W-A	SDR (dB)	SDRi (dB)
WASE (3, 8)	32-32	13.75	12.82
TinyWASE (3, 8)	4-8	13.86	12.93
TinyWASE (3, 8)	3-8	13.71	12.78
WASE (3, 4)	32-32	11.00	10.07
TinyWASE (3, 4)	4-8	10.80	9.87
TinyWASE (3, 4)	3-8	10.33	9.40

5.6. Ablation study

According to the experimental results mentioned above, we conduct an ablation study on Distillation-aware Quantization without the parameter sharing strategy based on the 3 × 8 TinyWASE model.

5.6.1. Effect of layer-wise quantization

We adopt layer-wise quantization in this paper because the weight distributions of the full-precision networks are quite different from layer to layer, as shown in Fig. 3. Table 7 compares the performance of shared quantization and layer-wise quantization. According to Table 7, layer-wise quantization shows superiority over shared quantization in our experiments.

5.6.2. Effect of non-uniform quantization

As shown in Fig. 3, the weight distributions of convolutional layers are subject to Gaussian distribution. Therefore, instead of using the commonly used linear quantization, in which the quantization intervals are equal, we adopt a non-uniform quantization strategy. The quantization intervals, which are the biases b_i in the quantization function Eq. (7), are initialized as the centroids of K-Means clustering over the weights, respectively. Table 8 shows that non-uniform quantization performs much better than linear quantization.

Table 7
Ablation study of training the quantization networks with shared quantization and layer-wise quantization. "W" and "A" represent the quantization bits of the weights and the activations, respectively.

Quantization methods	W-A	SDR (dB)	SDRi (dB)
Shared	3-8	14.70	14.57
Layer-wise	3-8	16.18	16.00

Table 8
Ablation study of training the quantization networks with linear quantization and non-uniform quantization. "W" and "A" represent the quantization bits of the weights and the activations, respectively.

Quantization methods	W-A	SDR (dB)	SDRi (dB)
Linear	3-8	15.65	15.52
Non-uniform	3-8	16.18	16.00

Table 9
Ablation study of training the quantization networks from scratch and from pretrained model. "W" and "A" represent the quantization bits of the weights and the activations, respectively.

Quantization methods	W-A	SDR (dB)	SDRi (dB)
From scratch	3-8	9.41	7.28
From pretrained	3-8	16.18	16.00

5.6.3. Effect of training from pretrained model

In our training, we train the quantized model from the well converged pretrained full-precision model for 50 epochs and linearly increase the temperature with respect to epoch ($T = 10 \times epoch$) with a learning rate of $5e^{-4}$. To investigate the effect of training from pretrained model, we also train a quantized model from scratch. We set the initial learning rate to $1e^{-3}$ for faster convergence and train for 50 epochs, then reduce the learning rate to $5e^{-4}$ and continue training for another 50 epochs. The temperature is set to $T = 5 \times epoch$ so that at the end of the experiment, both the model initialized from the pretrained full-precision model and the model trained from scratch reach the same temperature. As shown in Table 9, the performance of TinyWASE training from pretrained full-precision model shows significant superiority over the performance of TinyWASE training from scratch.

6. Conclusions

In this paper, we attempt to quantize the speaker extraction network for resource-constrained devices and propose Distillation-aware Quantization, which combines quantization and knowledge distillation techniques. We perform experiments on our previously proposed speaker extraction model WASE and get the quantized version TinyWASE. Our experiments give promising results, showing that the ultra-low bit TinyWASE can achieve comparable performance compared with full-precision WASE without much performance degradation. We further combine Distillation-aware Quantization with parameter sharing strategy, to achieve a larger compression ratio with some performance degradation. It is worth noting that it is possible to combine our proposed Distillation-aware Quantization with other model compression methods. Distillation-aware Quantization is simple and easy to implement and train, which is not constrained to our previously proposed WASE model. It is also possible to apply this method to other speaker extraction models, providing a novel solution towards on-device speaker extraction inference.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by the National Key Research and Development Program of China (2018AAA0100400), and the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB32070000).

References

Afouras, T., Chung, J. S., & Zisserman, A. (2019). My lips are concealed: Audio-visual speech enhancement through obstructions. In *INTERSPEECH*.

Ahn, S., Hu, S. X., Damianou, A., Lawrence, N. D., & Dai, Z. (2019). Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9163–9171).

Borgström, B. J., Brandstein, M. S., Ciccarelli, G. A., Quatieri, T. F., & Smalt, C. J. (2021). Speaker separation in realistic noise environments with applications to a cognitively-controlled hearing aid. *Neural Networks*, *140*, 136–147.

Bronkhorst, A. W. (2000). The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions. *Acta Acustica United with Acustica*, *86*(1), 117–128.

Chen, H., Du, J., Hu, Y., Dai, L.-R., Yin, B.-C., & Lee, C.-H. (2021). Correlating subword articulation with lip shapes for embedding aware audio-visual speech enhancement. *Neural Networks*.

Chen, X., Liu, G., Shi, J., Xu, J., & Xu, B. (2018). Distilled binary neural network for monaural speech separation. In *2018 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Chen, H., & Zhang, P. (2021). A dual-stream deep attractor network with multi-domain learning for speech dereverberation and separation. *Neural Networks*, *141*, 238–248.

Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and with two ears. *The Journal of the Acoustical Society of America*, *25*(5), 975–979.

Chung, S.-W., Choe, S., Chung, J. S., & Kang, H.-G. (2020). FaceFilter: Audio-visual speech separation using still images. In *INTERSPEECH* (pp. 3481–3485).

Cooke, M., Barker, J., Cunningham, S., & Shao, X. (2006). An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, *120*(5), 2421–2424.

Courbariaux, M., Bengio, Y., & David, J.-P. (2015). BinaryConnect: training deep neural networks with binary weights during propagations. In *Proceedings of the 28th international conference on neural information processing systems-Volume 2* (pp. 3123–3131).

Delcroix, M., Zmolikova, K., Kinoshita, K., Ogawa, A., & Nakatani, T. (2018). Single channel target speaker extraction and recognition with speaker beam. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5554–5558). IEEE.

Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., et al. (2018). Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. *ACM Transactions on Graphics*, *37*(4), 1–11.

Gu, R., Chen, L., Zhang, S.-X., Zheng, J., Xu, Y., Yu, M., et al. (2019). Neural spatial filter: Target speaker speech separation assisted with directional information. In *INTERSPEECH* (pp. 4290–4294).

Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International conference on learning representations*.

Hao, Y., Xu, J., Zhang, P., & Xu, B. (2021). Wase: Learning when to attend for speaker extraction in cocktail party environments. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 6104–6108). IEEE.

He, Y., Zhang, X., & Sun, J. (2017). Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 1389–1397).

Hershey, J. R., Chen, Z., Le Roux, J., & Watanabe, S. (2016). Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 31–35).

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. In *Deep learning workshop at the 28th international conference on neural information processing*.

Hou, L., & Kwok, J. T. (2018). Loss-aware weight quantization of deep networks. In *International conference on learning representations*.

Hu, R., Zhou, S., Tang, Z. R., Chang, S., Huang, Q., Liu, Y., et al. (2021). DMMAN: A two-stage audio–visual fusion framework for sound separation and event localization. *Neural Networks*, *133*, 229–239.

Huang, Z., & Wang, N. (2017). Like what you like: Knowledge distill via neuron selectivity transfer. arXiv preprint arXiv:1707.01219.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International conference on learning representations*.

Krishnamoorthi, R. (2018). Quantizing deep convolutional networks for efficient inference: A whitepaper. arXiv preprint arXiv:1806.08342.

Leng, C., Dou, Z., Li, H., Zhu, S., & Jin, R. (2018). Extremely low bit neural network: Squeeze the last bit out with admm. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 32*.

Li, C., Xu, J., Mesgarani, N., & Xu, B. (2021). Speaker and direction inferred dual-channel speech separation. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5779–5783). IEEE.

Li, X., Xu, Y., Yu, M., Zhang, S.-X., Xu, J., Xu, B., et al. (2021). MIMO self-attentive RNN beamformer for multi-speaker speech separation. In *INTERSPEECH*.

Luo, Y., Han, C., & Mesgarani, N. (2021). Ultra-lightweight speech separation via group communication. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 16–20). IEEE.

Michelsanti, D., Tan, Z.-H., Zhang, S.-X., Xu, Y., Yu, M., Yu, D., et al. (2021). An overview of deep-learning-based audio-visual speech enhancement and separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Polino, A., Pascanu, R., & Alistarh, D. (2018). Model compression via distillation and quantization. In *International conference on learning representations*.

Rastegari, M., Ordonez, V., Redmon, J., & Farhadi, A. (2016). Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision* (pp. 525–542). Springer.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. (2014). Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550.

Shamma, S. A., Elhilali, M., & Micheyl, C. (2011). Temporal coherence and attention in auditory scene analysis. *Trends in Neurosciences*, *34*(3), 114–123.

Szabó, B. T., Denham, S. L., & Winkler, I. (2016). Computational models of auditory scene analysis: a review. *Frontiers in Neuroscience*, *10*, 524.

Tuan, C.-I., Wu, Y.-K., Lee, H.-y., & Tsao, Y. (2019). Mitas: A compressed time-domain audio separation network with parameter sharing. arXiv preprint arXiv:1912.03884.

Wang, D., & Chen, J. (2018). Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(10), 1702–1726.

Wang, Q., Moreno, I. L., Saglam, M., Wilson, K., Chiao, A., Liu, R., et al. (2020). VoiceFilter-lite: Streaming targeted voice separation for on-device speech recognition. In *INTERSPEECH*.

Wang, Q., Muckenhirn, H., Wilson, K., Sridhar, P., Wu, Z., Hershey, J. R., et al. (2019). VoiceFilter: Targeted voice separation by speaker-conditioned spectrogram masking. In *INTERSPEECH* (pp. 2728–2732).

Wu, S., Li, G., Chen, F., & Shi, L. (2018). Training and inference with integers in deep neural networks. In *International conference on learning representations*.

Xu, C., Rao, W., Chng, E. S., & Li, H. (2020). SpEx: Multi-scale time domain speaker extraction network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 1370–1384.

Yang, J., Shen, X., Xing, J., Tian, X., Li, H., Deng, B., et al. (2019). Quantization networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7308–7316).

Yu, D., Kolbæk, M., Tan, Z.-H., & Jensen, J. (2017). Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 241–245). IEEE.

Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., & Zou, Y. (2016). Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160.

Žmolíková, K., Delcroix, M., Kinoshita, K., Ochiai, T., Nakatani, T., Burget, L., et al. (2019). SpeakerBeam: Speaker aware neural network for target speaker extraction in speech mixtures. *IEEE Journal of Selected Topics in Signal Processing*, *13*(4), 800–814.