IEEE Open Journal of Signal Processing

Received 11 April 2024; revised 19 June 2024; accepted 21 June 2024. Date of publication 9 July 2024;
date of current version 9 August 2024. The review of this article was arranged by Associate Editor Atsunori Ogawa.

Digital Object Identifier 10.1109/OJSP.2024.3425287

Short Paper

Fully Quantized Neural Networks for Audio Source Separation

ELAD COHEN, HAI VICTOR HABI (Graduate Student Member, IEEE), REUVEN PERETZ, AND ARNON NETZER

Sony Semiconductor Israel, Hod Hasharon 4524079, Israel

CORRESPONDING AUTHORS: REUVEN PERETZ; ARNON NETZER (email: reuven.peretz@sony.com; arnon.netzer@sony.com).

ABSTRACT Deep neural networks have shown state-of-the-art results in audio source separation tasks in recent years. However, deploying such networks, especially on edge devices, is challenging due to memory and computation requirements. In this work, we focus on quantization, a leading approach for addressing these challenges. We start with a theoretical and empirical analysis of the signal-to-distortion ratio (SDR) in the presence of quantization noise, which presents a fundamental limitation in audio source separation tasks. These analyses show that quantization noise mainly affects performance when the model produces high SDRs. We empirically validate the theoretical insights and illustrate them on audio source separation models. In addition, the empirical analysis shows a high sensitivity to activations quantization, especially to the network's input and output signals. Following the analysis, we propose Fully Quantized Source Separation (FQSS), a quantization-aware training (QAT) method for audio source separation tasks. FQSS introduces a novel loss function based on knowledge distillation that considers quantization-sensitive samples during training and handles the quantization noise of the input and output signals. We validate the efficiency of our method in both time and frequency domains. Finally, we apply FQSS to several architectures (CNNs, LSTMs, and Transformers) and show negligible degradation compared to the full-precision baseline models.

INDEX TERMS Source separation, quantization, DNN, SDR, compression, knowledge distillation.

I. INTRODUCTION

Recently, deep neural networks have shown great results on many audio tasks such as speech enhancement [1], [2], source separation [1], [2], [3], [4], [5], speech recognition [6], [7] and text-to-speech [8], [9]. However, such networks generally require a large memory footprint and high computational complexity, which makes their deployment on edge devices very challenging. Several approaches have been suggested to address these limitations, such as pruning [10], low-rank approximation [11], and quantization [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. The focus of this work is quantization, one of the most promising methods to overcome these challenges.

Quantization methods can be roughly divided into two categories: quantization-aware training (QAT) [13], [15], [16] and post-training quantization (PTQ) [14], [17]. PTQ is usually a simple technique that requires a small dataset to gather statistical information about the model. In contrast, QAT typically involves retraining the entire model on the full training dataset to compensate for the errors introduced by quantization. Although PTQ typically requires less computational resources, it often comes with sacrificing the accuracy compared to QAT.

In this work, we focus on QAT to harness its enhanced performance potential. The delivery of fully quantized low-precision models could be hardware-friendly, especially for edge devices, since all operations (e.g. addition, convolution, and matrix multiplication) are performed using low-precision integer arithmetic [15]. This allows the use of dedicated resource-constraint hardware to enable efficient deployment.

Several prior arts have quantized audio models in various ways. Low-bit quantization is suggested in [18], [19] for speech separation but is applied only to the model's weights to decrease its size. In [20], quantization is used only for convolutions, leaving other layers in floating-point operations. The authors of [22] use integer-only quantization, but it is specific for speech recognition. Recently, a new QAT method called Fully Quantized Speech Enhancement (FQSE) [23] suggested a new scheme for full 8-bit quantization. However, it was applied only to time-domain CNN-based models for speech enhancement. In addition, FQSE suggested data augmentation, which is used during QAT to strengthen the robustness of the quantized model. Nevertheless, the augmentation was specifically designed for speech enhancement tasks. This limits the extendability of FQSE to additional tasks since it requires a different augmentation

for each task, which can be nontrivial in some cases. As far as we know, there is an open gap in the capability to obtain fully quantized models for audio source separation tasks.

This paper aims to close this gap and obtain fully quantized models for audio source separation tasks that can be easily deployed on edge devices. We begin with a theoretical analysis of the signal-to-distortion ratio (SDR) [24] of quantized separation models, showing that quantization has a significant effect when the floating-point model produces high-performance results. The theoretical analysis is followed by an empirical one in which we first validate the theoretical results and then analyze which part of the quantized model contributed to the degradation in performance. Specifically, we show that input and output activations quantization significantly affect performance degradation. Based on our theoretical and empirical analysis, we propose a new QAT method for audio source separation tasks called Fully Quantized Source Separation (FQSS), which enables 8-bit full quantization for efficient deployment. FQSS utilizes knowledge distillation [25] in a unique way, which guides the training process to reduce the quantization error, especially in cases where quantization causes considerable degradation in performance. In addition, we correct the input and output quantization errors by extending the quantization scheme from [23] to speech and music separation, as well as to frequency domain separation approaches. Finally, we conduct several experiments to validate our method on various architectures (e.g., CNNs, LSTMs, and Transformers) over speech and music separation tasks. FQSS shows a significant improvement in memory footprint and the number of bit operations while having a negligible degradation compared to the floating-point model.

Our contributions are summarized as follows:

- We present a theoretical and empirical analysis of the SDR for audio source separation in the presence of quantization noise.
- We introduce a new QAT method for speech and music separation, using a novel knowledge distillation loss that pays attention to the quantization-sensitive samples in the retraining process.
- We present results on several architectures such as CNNs (Conv-TasNet [2]), LSTMs (DPTNet [3]) and Transformers (Sepformer [4], HTDemucs [5]) and validate their performance using 8-bit full quantization for both weights and activations.

In the spirit of reproducible research, our code is available at https://github.com/ssi-research/FQSS.

II. BACKGROUND

This section provides essential background such as quantization and knowledge distillation, along with the splitter and reconstructor scheme from [23], which are fundamental for understanding the method described in our work.

A. QUANTIZATION

Here, we provide a short overview of quantization. A uniform quantizer maps a floating-point number to its nearest quantization level. Given a scalar x with quantizer step-size Δ, zero-point z, and bit-width b, the quantization of x is represented by:

$$\mathrm{Q}(x) \triangleq \Delta \cdot \mathrm{clip}\left(\left\lfloor \frac{x-z}{\Delta} \right\rceil, 0, 2^b - 1\right) + z, \tag{1}$$

where $\mathrm{clip}(x, \alpha, \beta)$ limits x to the range between α and β, and $\lfloor \cdot \rceil : \mathbb{R} \to \mathbb{Z}$ is the rounding operation. This formula is used element-wise for a vector x. There are various ways to select the quantization parameters Δ and z. In this work, we use min-max, similarly to [15]. For symmetric signed quantization, $z = -t$ and $\Delta = \frac{t}{2^{b-1}}$, where $t = \max_i(|\mathbf{x_i}|)$, and for asymmetric quantization $\Delta = \frac{\max_i(\mathbf{x_i}) - \min_i(\mathbf{x_i})}{2^b - 1}$ and $z = \min_i(\mathbf{x_i})$.

B. KNOWLEDGE DISTILLATION

Knowledge Distillation (KD) is a method for training compact yet efficient models using a high-capacity 'teacher' model to train a smaller 'student' model. This strategy has been widely applied in diverse fields, such as computer vision [17], speech processing [26], [27], etc. Several adaptations of KD have been investigated to train models dedicated to audio. [26] is a teacher-student technique that trains multiple teacher models for distinct signal-to-noise ratio (SNR) ranges. In [27], layer-wise learning and objective shifting mechanisms are used to distill knowledge from large Transformer-based speech separation models. In our work, we apply KD by having the 'teacher' model (the floating-point model) instruct the 'student' model (the quantized model), mainly focusing on quantization-sensitive samples.

C. SPLITTER AND RECONSTRUCTOR

FQSE [23] shows that a large part of the performance degradation was caused mainly by quantizing the model's input and output activation tensors. It overcomes this degradation by introducing the splitter and reconstructor which keeps the input and output waveforms at high-precision (16-bit) while only using low-precision operations (8-bit). Here, we briefly describe the splitter and reconstructor scheme.

The splitter splits a single input channel of 16-bit into two channels of 8-bit using bits-splitting. Specifically, denote the bit-width of a low-precision as b_l. The splitting operation can be expressed as: $\mathbf{X} = \begin{bmatrix} \mathrm{Q_F}(x) \\ \mathrm{Q_F}(\frac{2 \cdot t \cdot \epsilon}{\Delta} - t) \end{bmatrix}$ where x is the input sample and $\mathrm{Q_F}$ is a symmetric floor quantizer. ϵ is the residual error: $\epsilon \triangleq x - \mathrm{Q_F}(x)$ and the symmetric floor quantizer is given by: $\mathrm{Q_F}(x) \triangleq \Delta \cdot \mathrm{clip}(\lfloor \frac{x}{\Delta} \rfloor, -2^{b_l - 1}, 2^{b_l - 1} - 1)$, where $\lfloor \cdot \rfloor : \mathbb{R} \to \mathbb{Z}$ denotes the floor operation. To restrict the residual error in the same range $[-t, t)$ as $\mathrm{Q_F}(x)$, they scale it by $\frac{2 \cdot t}{\Delta}$ and subtract t. This way, $\mathbf{X}$ remains per tensor quantization. This split requires changing the splitter's first layer to support two channels (instead of one channel), the same as the input.

The reconstructor combines two outputs of 8-bit into one output of 16-bit. Specifically, let $\mathbf{Y} \in \mathbb{R}^{c \times k}$, $\mathbf{X} \in \mathbb{R}^{1 \times n}$ be the input (features) and output (samples) of the model's decoder, respectively, where c is the number of channels, k is the number of elements in each channel and n is the number of output samples. Then, FQSE introduces a Residual Quantization Block (RQB) to produce a quantization correction term. RQB consists of additional encoder $\widetilde{E} : \mathbb{R}^{1 \times n} \to \mathbb{R}^{c \times k}$ and additional decoder $\widetilde{D} : \mathbb{R}^{c \times k} \to \mathbb{R}^{1 \times n}$. It projects the low-precision samples back into the feature space using $\widetilde{E}$, which results in $\widetilde{\mathbf{Y}} = Q(\widetilde{E}(\mathbf{X}))$. Then, using the low-precision features, computes the residual error in feature space by $\mathbf{U} = Q(\mathbf{Y} - \widetilde{\mathbf{Y}})$. Finally, the error is projected back using $\widetilde{D}$, which results in the correction term $\boldsymbol{\delta}_{\boldsymbol{X}} = Q(\widetilde{D}(\mathbf{U}))$. The final quantized output is achieved by scaling and adding the correction term and given by $Q(\mathbf{X}) + \frac{\boldsymbol{\delta}_{\boldsymbol{X}}}{2^{b_l - 1}} \in \mathbb{R}^{1 \times n}$. Detailed RQB scheme is shown in [23]. Finally, the model is retrained with QAT, where the new parameters of the splitter and reconstructor are learned. Since quantization is a non-differentiability operation, a Straight-Through Estimator (STE) [28] is used to compute the gradients during the retraining. In our work, we extend the

splitter-reconstructor scheme to multi-source separation tasks and to frequency domain approaches.

III. ANALYSIS OF QUANTIZED MODELS

In this section, we analyze the effect of model quantization on the performance of source separation tasks. Specifically, we investigate the SDR [24] of quantized models. We present a theoretical analysis in which we provide an upper bound (Theorem III.1) for the performance of the quantized model in terms of the SDR of the floating-point model and the quantization noise. In addition, we validate Theorem III.1 on models trained using QAT and present empirical findings to support our formulation.

A. THEORETICAL ANALYSIS

We start by defining the SDR. Specifically, let $\boldsymbol{s}_m \in \mathbb{R}^n$ be the m-th clean source and $\boldsymbol{g} = \sum_{m=1}^{M} \boldsymbol{s}_m$ be a mixture of M sources where M is the number of sources. Given a floating-point model f_m that separates the m-th source, its output is defined as $\hat{\boldsymbol{s}}_m = f_m(\boldsymbol{g}) = \boldsymbol{s}_m + \boldsymbol{w}_m$ where $\boldsymbol{w}_m$ is the model noise. Then, the SDR [29] for multiple sources is defined as:

Definition III.1 (Signal-to-Distortion Ratio (SDR)):

$$\mathrm{SDR} \triangleq \frac{1}{M}\sum_{m=1}^{M} 10\log_{10}\frac{\mathbb{E}\left[\|\boldsymbol{s}_m\|_2^2\right]}{\mathbb{E}\left[\|\hat{\boldsymbol{s}}_m - \boldsymbol{s}_m\|_2^2\right]}, \tag{2}$$

where $\|\cdot\|_2$ denotes the L2 norm and $\mathbb{E}[\cdot]$ denotes the expectation w.r.t. all random variables.

Similarly, for a quantized model of the m-th source f_m^{Q}, we obtain $\tilde{\boldsymbol{s}}_m = f_m^{\mathrm{Q}}(\boldsymbol{g}) = \hat{\boldsymbol{s}}_m + \boldsymbol{v}_m$ where $\boldsymbol{v}_m$ is the noise due to the model quantization. Similar to (2), $\mathrm{SDR}_{\mathrm{Q}}$ is the SDR of the quantized model for multiple sources and is defined as:

Definition III.2 (Quantized SDR):

$$\mathrm{SDR}_{\mathrm{Q}} \triangleq \frac{1}{M}\sum_{m=1}^{M} 10\log_{10}\frac{\mathbb{E}\left[\|\boldsymbol{s}_m\|_2^2\right]}{\mathbb{E}\left[\|\tilde{\boldsymbol{s}}_m - \boldsymbol{s}_m\|_2^2\right]}. \tag{3}$$

We aim to understand the effect of the quantization noise on the performance of separation models. Thus, we provide an upper bound on the SDR of quantized models in the following Theorem:

Theorem III.1 (SDR of Quantized Models):

$$\mathrm{SDR}_{\mathrm{Q}} \leq \mathrm{SDR} - \frac{10}{M}\log_{10}\left(1 + 10^{M\cdot\frac{\mathrm{SDR}-\mathrm{SQNR}}{10}}\right) + \widetilde{\alpha}, \tag{4}$$

where $\widetilde{\alpha} = -\frac{10}{M}\log_{10}(1-\alpha_{\max})$, and $\alpha_{\max} = \max_m \left|\frac{\mathbb{E}[\boldsymbol{w}_m^T \boldsymbol{v}_m]}{\sqrt{\mathbb{E}[\|\boldsymbol{v}_m\|_2^2]}\sqrt{\mathbb{E}[\|\boldsymbol{w}_m\|_2^2]}}\right|$ is the maximal absolute normalized correlation factor between the model and quantization noises, and $\mathrm{SQNR} = \frac{1}{M}\sum_{m=1}^{M} 10\log_{10}\frac{\mathbb{E}[\|\boldsymbol{s}_m\|_2^2]}{\mathbb{E}[\|\boldsymbol{v}_m\|_2^2]}$ is the Signal-to-Quantization Noise Ratio.

A detailed proof of Theorem III.1 is given in Appendix A. From Theorem III.1, we observe that the SDR gap between the floating-point and the quantized models increases when the SDR of the floating-point model increases. This means that the effect of quantization is significant at high SDRs, while it has a minor effect at low SDRs.

In the following Corollary, we analyze the theoretical results by examining two corner cases. Note that by $a \ll b$ we mean that $\frac{b}{a} \to \infty$.

Corollary III.1.1 (Asymptotic Quantized SDR):

$$\mathrm{SDR}_{\mathrm{Q}} = \begin{cases} \mathrm{SDR} & \mathrm{SDR} \ll \mathrm{SQNR}, \\ \mathrm{SQNR} & \mathrm{SDR} \gg \mathrm{SQNR}. \end{cases} \tag{5}$$

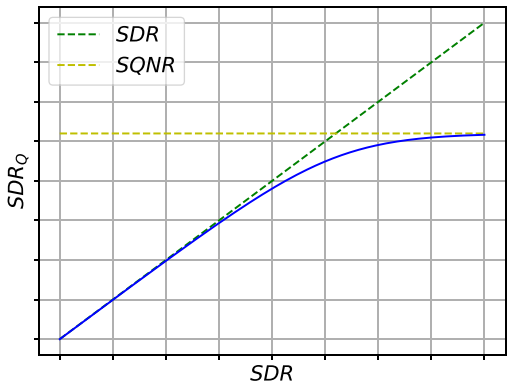


FIGURE 1. Asymptotic lines of SDR and SQNR from Corollary III.1.1 that illustrate the behavior of $\mathrm{SDR}_{\mathrm{Q}}$.

A detailed proof of Corollary III.1.1 is given in Appendix B. From (5), we observe two asymptotic behaviors. When SDR $\ll$ SQNR, the model noise is the dominant noise, and when SDR $\gg$ SQNR, the quantization noise is the dominant noise. This shows that $\mathrm{SDR}_{\mathrm{Q}}$ is mostly limited by SQNR at high SDRs. This is illustrated in Fig. 1, where we present asymptotic lines over a range of SDR values.

B. EMPIRICAL FINDINGS

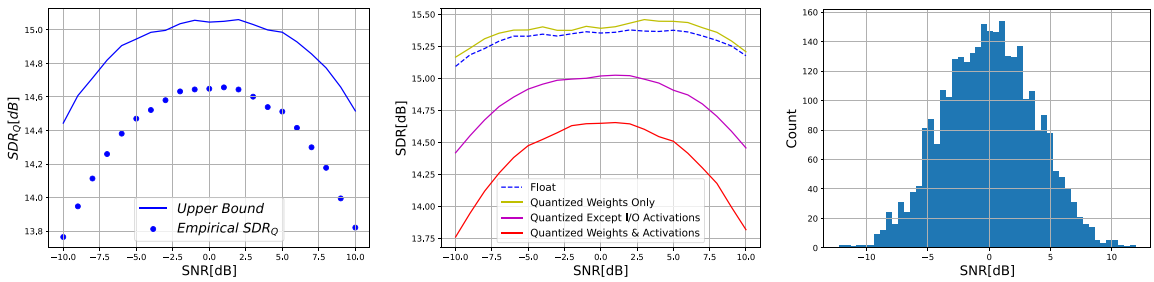
Following the theoretical results, we present empirical findings demonstrating Theorem III.1. We quantize Conv-TasNet [2] for the separation of two speakers using 8-bit vanilla QAT as described in [15], which is a common quantization practice. Throughout this paper, we refer to signal-to-noise ratio (SNR) as the signal-power ratio between the sources in the context of audio source separation tasks. Since other sources are considered "noise" for a specific source, the SNR (dB) of the m-th source is quantitatively measured by $\mathrm{SNR}_m \triangleq 10\log_{10}(\frac{\|\boldsymbol{s}_m\|_2^2}{\|\boldsymbol{g}-\boldsymbol{s}_m\|_2^2})$. The mixtures are created from the LibriMix [30] test set with a wide range of SNRs. Additional experimental details are described in section V.

Fig. 2(a) shows that empirical $\mathrm{SDR}_{\mathrm{Q}}$ values match the upper bound from Theorem III.1. The x-axis is the SNR of the first speaker, where high values represent high SNRs, which are equivalent to low SNRs for the second speaker. In addition, in Fig. 2(b), we compare the SDR [24] of the baseline floating-point model to the SDR of the quantized model where different parts of the model are quantized: 1) weights only[1]; 2) weights and activations; 3) weights and activations except for the input-output signals. It can be seen that weights quantization does not cause any performance degradation and when the activations are quantized, there is a considerable degradation in performance [21], [23]. We also observe that quantization mostly affects high SNR sources (i.e. -10 dB or 10 dB) and the most sensitive parts are the input and output. In such cases, quantization noise becomes more dominant, and its effect is significant.

IV. METHOD

We suggest *Fully Quantized Source Separation* (FQSS), a method to enable efficient deployment for audio source separation models. FQSS's quantization technique consists of two parts: 1) *SDR-Aware*

[1]Note that in QAT, depending on the bit-width and quantization method, performance might exceed the floating-point precision.



(a) Empirical SDR_Q vs. upper bound (b) Quantization impact on SDR (c) SNR histogram

FIGURE 2. Analysis of SDR under Quantization. (a) Empirical SDR_Q against the theoretical upper bound across various SNRs. (b) Impact of quantization on SDR, comparing the baseline floating-point model against a vanilla QAT model where different parts of the model are quantized. (c) A histogram of the SNR for two speakers using LibriMix [30] test set.

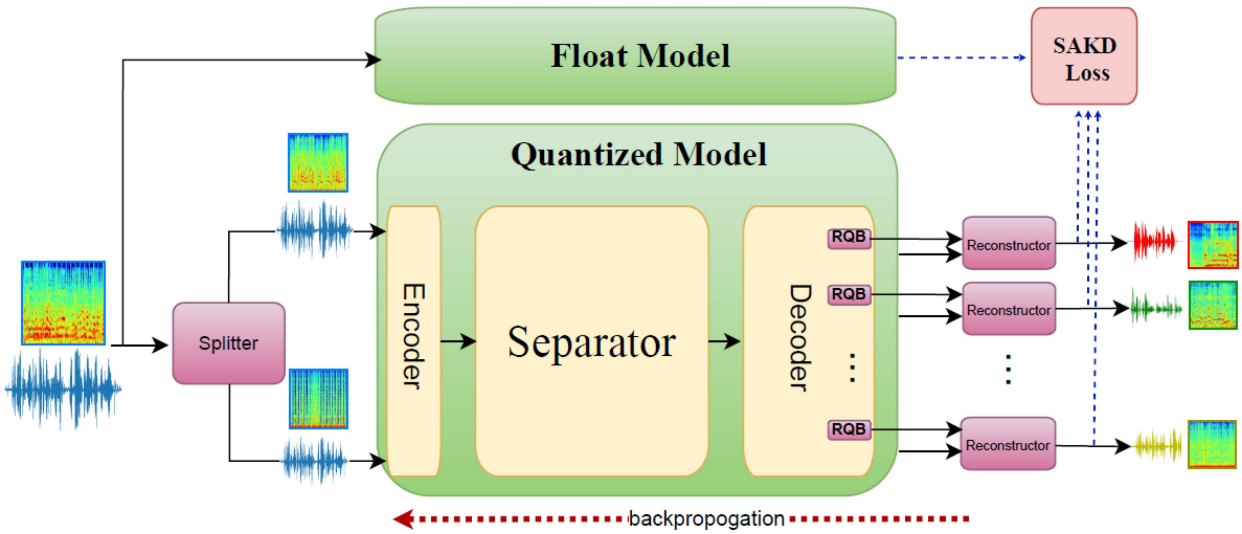


FIGURE 3. FQSS scheme. For the input, we split each high-precision channel into multiple low-precision. The Separator extracts per-source feature maps from the mixed features. For the output, multiple RQBs in the decoder extract error quantization terms to correct the model's low-precision outputs. Handling the input and output enables full quantization with only low-precision tensors. The floating-point model is used for QAT with SDR-Aware Knowledge Distillation (SAKD). Signals are waveforms (1D) or spectrograms (2D).

Knowledge Distillation (SAKD), which is a new loss that emphasizes quantization-sensitive samples by focusing on these samples while minimizing the overall task loss; 2) extending splitter-reconstructor to multi-source separation tasks as well as to frequency domain approaches. Fig. 3 illustrates the extended splitter-reconstructor and their interface with the input and output, respectively. This scheme tackles the primary source of the quantization error, which is the input and output quantization, as demonstrated in Fig. 2(b). The new scheme is fully quantized and trained with the SAKD loss function. We describe SAKD in Section IV-A and the extension of the splitter-reconstructor from the previous work [23] in Section IV-B.

A. SDR-AWARE KNOWLEDGE DISTILLATION

Based on Theorem III.1, quantization noise can cause performance drops, particularly at high SDRs, where the model noise is small and the quantization noise becomes more dominant. Paying attention to quantization-sensitive cases can improve the robustness of the quantized model. We suggest SDR-Aware Knowledge Distillation (SAKD) to handle samples with low SDR_Q detected during training. Knowledge distillation forces the quantized model to minimize the difference between its outputs and the floating-point model's outputs. Additional SDR information can guide the QAT learning process. Thus, we add a per-sample weighting to the task loss to consider data samples relative to their quantization errors. In this way, the QAT process can minimize the quantization effect, in particular when the SDR of the floating-point model is much better than the quantized SDR. Specifically, let us consider a floating-point model f^Θ, where Θ represents the parameters of the model, trained to minimize a task loss $\mathcal{L}_{task}$ on dataset $\mathcal{D}$. The dataset $\mathcal{D}$ consists of N pairs $(\boldsymbol{g}_i, \mathcal{S}_i)$ where $\boldsymbol{g}_i \in \mathbb{R}^{1\times n}$ is a mixture sample and $\mathcal{S}_i = \{\boldsymbol{s}_k \in \mathbb{R}^{1\times n}\}_{k=1}^M$ is a set of M ground truth clean sources. The quantized form of f^Θ is denoted as f^Ω, where Ω includes the quantized representation of Θ and the quantization parameters Δ, z of all quantizers. The m-th separated source of f^Θ and f^Ω are denoted by f_m^Θ and f_m^Ω, respectively. For the sample $\boldsymbol{g}_i$, we define its *quantization-sensitivity* weighting by:

$$\gamma_i = 10^{\left(\frac{\text{SDR}_i - \text{SDR}_{\text{Q}i}}{10}\right)}, \tag{6}$$

where $\text{SDR}_i \triangleq \frac{10}{M}\sum_{m=1}^M \log_{10}\frac{\|\boldsymbol{s}_m\|_2^2}{\|f_m^\Theta(\boldsymbol{g}_i)-\boldsymbol{s}_m\|_2^2}$ and $\text{SDR}_{\text{Q}i} \triangleq \frac{10}{M}\sum_{m=1}^M \log_{10}\frac{\|\boldsymbol{s}_m\|_2^2}{\|f_m^\Omega(\boldsymbol{g}_i)-\boldsymbol{s}_m\|_2^2}$ are the SDRs (dB) of the floating-point

and quantized models (f^{Θ} and f^{Ω}, respectively), for the sample $\boldsymbol{g}_i$. The overall loss used to train the quantized model is defined as follows:

$$\mathcal{L}_{total}(\Omega) = (1-\lambda)\sum_{i=1}^{N}\mathcal{L}_{task}\left(f^{\Omega}(\boldsymbol{g}_i), S_i\right) + \lambda\sum_{i=1}^{N}\gamma_i\cdot\mathcal{L}_{kd}\left(f^{\Omega}(\boldsymbol{g}_i), f^{\Theta}(\boldsymbol{g}_i)\right), \quad (7)$$

where λ is a hyper-parameter to weigh the knowledge distillation loss term, and $\mathcal{L}_{kd}$ is the knowledge distillation loss between the floating-point model and quantized model outputs. The $\mathcal{L}_{kd}$ focuses on improving the SQNR by addressing only the quantization noise and not the model noise that is added in both models. This loss, supported by our theoretical analysis, increases the potential for improving SDR_{Q} in quantization-sensitive SNR ranges and the overall performance of the quantized model.

B. SPLITTER AND RECONSTRUCTOR

In [23], the Splitter and RQB components are designed to reduce quantization noise in single-source, time-domain models, as discussed in Section II. This work extends its functionality in two significant directions:

Multiple Sources: We adapt the RQB to support multiple sources, enabling its usefulness in source separation tasks. The RQB learns to produce a quantization correction term which is the quantization error. Since the quantization errors do not depend on a specific source, we share the RQB's parameters across all sources. Specifically, let $R_{\widetilde{E},\widetilde{D}} : (\mathbf{Y}, \mathbf{X}) \rightarrow \boldsymbol{\delta}_X$ be the RQB where $\mathbf{Y}$ and $\mathbf{X}$ are the input (features) and output (samples) of the model's decoder, respectively, and $\boldsymbol{\delta}_X$ is the residual quantization error. In the case of multiple sources, we denote $\mathbf{Y_i}$ and $\mathbf{X_i}$ as the input and output of the model's decoder for source i, respectively. Then, the RQB's parameters $(\widetilde{E}, \widetilde{D})$ can be reused by applying $R_{\widetilde{E},\widetilde{D}}(\boldsymbol{Y}_i, \boldsymbol{X}_i) = \boldsymbol{\delta}_{\boldsymbol{X}_i} \quad \forall i \in \{1, 2, \ldots, M\}$. We repetitively use each correction term $\boldsymbol{\delta}_{\boldsymbol{X}_i}$ to correct each separated source $\boldsymbol{X}_i$ by applying $Q(\boldsymbol{X}_i) + \frac{\delta_{\boldsymbol{X}_i}}{2^{b_l-1}}$.

Frequency Domain: We extend the splitter-reconstructor applicability to include frequency domain approaches, where the inputs are spectrograms. The real and imaginary components of the spectrograms are folded into the channel dimension. Then, we apply a splitter to replace a single high-precision spectrogram with two low-precision spectrograms. We add weights to the model's encoder to align the new input channel count, initializing them with a Gaussian distribution matching the existing weights' mean and variance. In this context, the RQB corrects quantization errors in the frequency domain. This results in spectrograms that are 8-bit fully quantized.

V. EXPERIMENTS

In this section, we present experiments for two tasks: speech separation and music separation. In all of our experiments, we aligned $\mathcal{L}_{kd}$ to be equivalent to $\mathcal{L}_{task}$ and set $\lambda = 0.1$ unless otherwise noted. We use per-channel symmetric signed quantization for the weights and asymmetric per-tensor quantization for activations. For each model, we report performance metrics, size, and Tera Bit-Operations (TBOPs). BOP is the total number of bit operations that is hardware-agnostic and resembles the model's computational complexity. TBOPs are evaluated for a 1-second audio segment. In addition, we present an ablation study to demonstrate our method on a wide range of SNRs. Appendix C describes Algorithm 1 for simulating SNR levels. For simplicity, we denote the splitter and reconstructor for the input and output as IN and OUT, respectively. Unless mentioned otherwise, all models were trained using the Adam [31] optimizer on 4 NVIDIA V100 32 GB GPUs.

A. IMPLEMENTATION DETAILS

1) SPEECH SEPARATION

Training: We use the LibriMix [30] dataset for retraining and testing our method for the separation of two speakers. LibriMix is derived from the LibriSpeech [32] data set and the WHAM [33] noises. It consists of three splits: train, dev, and test. Each split contains short mixed/noisy speech, 16 bits per sample at 8 kHz. For training, we use the train split (train-360), which contains 50,800 samples. We use the shortest waveform length between the mixed and clean signals. The loss function is negative SI-SDR [24]. Conv-TasNet [2] was trained with a learning rate of 10^{-3} and batch size of 6 and we use the same training recipe as in [34]. DPTNet and Sepformer were trained with a batch size of 1 and learning rates of $4 \cdot 10^{-4}$ and $1.5 \cdot 10^{-4}$ following the settings in [3] and [4], respectively. In DPTNet, internal activations in LSTM layers were quantized by dynamic quantization [12].

Evaluation: We use the LibriMix test split, which contains 3,000 samples. The following metrics are reported: SI-SDR, SDR [24], and Short-Time Objective Intelligibility Measure (STOI) [35]. We use TorchMetrics [36] for the implementation of the metrics. Furthermore, we present SI-SDR results in several SNRs to emphasize the increased sensitivity of quantization and the advantages of FQSS in such scenarios. Specifically, we select three SNR values: low (-10 dB), medium (0 dB), and high (10 dB), to represent a range of the SNRs observed in Fig. 2(c). For each selected SNR, we simulate the entire test set using Algorithm 1.

2) MUSIC SEPARATION

Training: We use the Musdb18HQ [37] data set to retrain and test our method. Musdb18HQ is a high-quality version of the Musdb18 [38] dataset, which consists of 150 songs sampled at 44.1 kHz, 100 of them for training, provided as raw WAV files. We apply FQSS on Conv-TasNet [2] and HTDemucs [5], which is a hybrid (time and frequency) transformers-based model. In HTDemucs, we evaluate our method on both time and frequency networks. We follow the training setups of [39] and [5] with batch size of 32 and learning rates of 10^{-5} and $3 \cdot 10^{-4}$ for Conv-TasNet and HTDemucs, respectively. The loss function is an L1 norm loss. Additionally, HTDemucs was trained on 8 NVIDIA A100 80 GB GPUs.

Evaluation: We use the Musdb18HQ test set, which contains 50 songs. We report the total SDR and the SDR per instrument as defined by the SiSEC18 [40], the median across the median SDR over all 1-second chunks in each song. Similar to speech separation, we evaluate the separation of vocals, where all other instruments act as noise, in three values of SNR: low, medium, and high, which are -15 dB, -5 dB, and 5 dB, respectively. These values have been selected to cover SNR conditions of input vocals based on Musdb18HQ histogram, and we simulate the entire test set for each SNR. Here, we skip audio segments without vocals.

B. RESULTS

Speech Separation: Table 1 compares FQSS's results with floating-point and vanilla QAT. We show that our approach reduces sensitivities across a wide range of SNRs. FQSS outperforms vanilla QAT on the LibriMix test. In particular, it has the most SI-SDR improvements when one speaker is much stronger than the other. Our 8-bit quantized model cuts the size of the floating-point model by 4x

TABLE 1. Speech Separation. Comparison Between Floating-Point Models and Their 8-Bit Fully Quantized Models on LibriMix [30]

Model	Precision	Model Size [MB]	TBOPs	SI-SDR[dB]			SI-SDR [dB]	SDR [dB]	STOI
				Low	Mid	High			
Conv-TasNet	Float	20.30	5.136	14.55	14.82	14.64	14.78	15.31	0.931
	Vanilla QAT	5.20	0.321	13.28	14.15	13.34	14.23	14.75	0.916
	FQSS (Ours)	5.22	0.324	14.42	14.76	14.49	14.75	15.20	0.928
DPTNet	Float	11.40	57.216	18.27	18.37	18.25	18.50	19.03	0.957
	Vanilla QAT	2.95	3.576	16.89	17.51	16.58	17.88	17.97	0.951
	FQSS (Ours)	2.97	3.577	18.32	18.30	18.28	18.30	18.80	0.956
Sepformer	Float	113.10	70.448	19.12	19.18	19.11	19.17	19.64	0.963
	Vanilla QAT	28.55	4.403	18.43	18.21	17.47	18.35	18.74	0.951
	FQSS (Ours)	28.57	4.407	18.49	18.78	18.57	18.91	19.31	0.959

Comparison between floating-point models and their 8-bit fully quantized models on LibriMix [30]. The separation is for a mixture of two speakers.

TABLE 2. Music Separation. Comparison Between Floating-Point Models and Their 8-Bit Fully Quantized Models on Musdb18HQ [37]

Model	Precision	Model Size [MB]	TBOPs	Vocals SDR[dB]			SDR[dB]				
				Low	Mid	High	Drums	Bass	Other	Vocals	Total
Conv-TasNet	Float	44.00	49.60	0.98	6.17	12.44	6.12	6.37	4.40	6.63	5.88
	Vanilla QAT	11.20	3.10	0.92	5.34	9.49	5.93	5.84	4.42	6.42	5.65
	FQSS (Ours)	11.22	3.57	1.13	6.64	12.72	5.92	6.35	4.43	6.59	5.82
HTDemucs	Float	168.10	185.28	1.69	7.60	14.11	8.15	8.61	5.43	7.87	7.51
	Vanilla QAT	42.25	11.58	1.30	6.73	12.14	7.58	8.37	4.90	6.47	6.83
	FQSS (Ours)	42.27	11.77	1.67	7.30	12.76	7.90	8.85	5.32	7.67	7.43

Comparison between floating-point models and their 8-bit fully quantized models on Musdb18HQ [37]. The separation is for stereophonic audio consisting of Drums, Bass, Vocals, and Other.

and the complexity by 16x while keeping similar performance. The overhead of the splitter and reconstructor is minimal due to simple bit-splitting and addition.

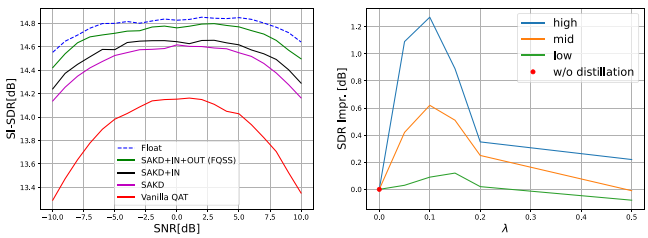
Music Separation: Table 2 compares the performance of FQSS with the floating-point and vanilla QAT. FQSS outperforms vanilla QAT on the Musdb18HQ test set. Specifically, it improves the total SDR by 0.6 dB for HTDemucs [5]. Results suggest that 16-bit precision can be replaced with 8-bit without sacrificing performance for the time and frequency domains. Our model is 4x smaller and 16x faster than the floating-point model, which is crucial for edge devices. This is with a negligible performance degradation compared to the floating-point baseline.

C. ABLATION STUDY

Here, we provide an ablation study for our proposed method. First, we evaluate FQSS method incrementally to show the effectiveness of each component. Then, we explore several values of λ for SAKD. Finally, we present an ablation for the splitter-reconstructor on a frequency domain network.

Incremental Ablation: To investigate the contribution of FQSS components, we present incremental ablation for speech separation using Conv-TasNet[2] [2]. We begin with a vanilla QAT and gradually add FQSS components as follows: SAKD, SAKD+IN, and FQSS (SAKD+IN+OUT). We present the SI-SDR results in Fig. 4(a). It illustrates that SAKD plays a significant role in improving performance. The results show an improvement across a wide range of mixed speech and especially high SNRs, where we reach an improvement of more than 1 dB compared to vanilla QAT. This ablation shows the impact of each component, showing performance drops when a component is absent.

[2]Similar results were observed with both speech and music separation, as well as with other models.



(a) Incremental ablation (b) SAKD

FIGURE 4. Ablation. (a) Incremental ablation for FQSS. (b) SDR improvement of the vocals separation for various distillations (λ) in SAKD. High SNRs show the most improvement.

TABLE 3. Frequency Domain Ablation. Evaluating SDR on HTDemucs [5] for Different Input and Output Precisions

Input	16-bit	IN	16-bit	IN	8-bit	8-bit
Output	16-bit	16-bit	OUT	OUT	16-bit	8-bit
SDR[dB]	7.42	7.42	7.43	7.43	7.31	7.13

Evaluating SDR on HTDemucs [5] for different input and output precisions. IN and OUT are the splitter and reconstructor, respectively.

SAKD: Fig. 4(b) provides an ablation for SAKD using Conv-TasNet [2] for music separation[2]. Here, we set several values to the hyperparameter λ, starting from $\lambda = 0$ (w/o distillation), and train a quantized model for each value. Then, we evaluate the vocals separation for the same three SNR levels: low, mid, and high. The figure shows a significant improvement around $\lambda = 0.1$ compared to the results without distillation. The knowledge obtained from the floating-point model can boost performance, especially at high SNRs, while having a minor effect at low SNRs.

Frequency Domain Network: Table 3 shows an ablation for the frequency domain network in HTDemucs [5]. Here, we keep the input-output of the time domain network in 16-bit while using the splitter-reconstructor scheme only in the frequency domain. The table indicates the effectiveness of the splitter-reconstructor in avoiding performance loss caused by 8-bit input-output quantization for spectrograms.

VI. CONCLUSION

In this paper, we developed a theoretical upper bound for SDR in quantized models, showing a particular effect on high-performing floating-point models. Our empirical analysis highlights the quantized model's sensitivities, especially when the input and output are quantized. We proposed a new QAT method for audio source separation, using a novel knowledge distillation loss that focuses on sensitive samples during the retraining. In addition, we extend the FQSE's splitter-reconstructor to source separation tasks. Our method, FQSS, shows model size and complexity reductions with minor performance drops on various architectures and tasks. This work presents the first fully quantized models for audio separation, which paves the way for future research in further directions, such as PTQ, mixed precision, and beyond.

APPENDIX A
THEORETICAL ANALYSIS OF QUANTIZED MODELS

A. PROOF: SDR OF QUANTIZED MODELS

Proof: First, we look at the log argument in (3):

$$\psi = \prod_{m=1}^{M} \frac{P_{\boldsymbol{s}}^{(m)}}{P_{\boldsymbol{v}}^{(m)} + P_{\boldsymbol{w}}^{(m)} + 2\mathbb{E}\left[\boldsymbol{w}_m^T \boldsymbol{v}_m\right]}, \tag{8}$$

where $P_{\boldsymbol{s}}^{(m)} = \mathbb{E}[\|\boldsymbol{s}_m\|_2^2]$, $P_{\boldsymbol{v}}^{(m)} = \mathbb{E}[\|\boldsymbol{v}_m\|_2^2]$ and $P_{\boldsymbol{w}}^{(m)} = \mathbb{E}[\|\boldsymbol{w}_m\|_2^2]$ are the powers of the vectors $\boldsymbol{s}_m$, $\boldsymbol{v}_m$ and $\boldsymbol{w}_m$, receptively. Now, we investigate the denominator of (8):

$$\begin{aligned}
\beta_m &\triangleq P_{\boldsymbol{v}}^{(m)} + P_{\boldsymbol{w}}^{(m)} + 2\mathbb{E}\left[\boldsymbol{w}_m^T \boldsymbol{v}_m\right] \\
&= P_{\boldsymbol{w}}^{(m)} \left(1 + \frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}}\right) \left(1 + \frac{2\mathbb{E}\left[\boldsymbol{w}_m^T \boldsymbol{v}_m\right]}{\left(1 + \frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}}\right) P_{\boldsymbol{w}}^{(m)}}\right) \\
&\geq P_{\boldsymbol{w}}^{(m)} \left(1 + \frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}}\right) \left(1 - \frac{2\alpha_{\max}\sqrt{\frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}}}}{\left(1 + \frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}}\right)}\right),
\end{aligned} \tag{9}$$

where $\alpha_{\max} = \max_m \frac{|\mathbb{E}[\boldsymbol{w}_m^T \boldsymbol{v}_m]|}{\sqrt{\mathbb{E}[\|\boldsymbol{v}_m\|_2^2]}\sqrt{\mathbb{E}[\|\boldsymbol{w}_m\|_2^2]}}$ is the maximal absolute normalized correlation factor. For the following step, let's define $\boldsymbol{w}_m = \gamma_{\boldsymbol{w}}^{(m)} \bar{\boldsymbol{w}}_m$ and $\boldsymbol{v}_m = \gamma_{\boldsymbol{v}}^{(m)} \bar{\boldsymbol{v}}_m$ such that $\mathbb{E}[\|\bar{\boldsymbol{w}}_m\|_2^2] = 1$ and $\mathbb{E}[\|\bar{\boldsymbol{v}}_m\|_2^2] = 1$. Writing the following term w.r.t. $u_m \triangleq \frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}} = \frac{\mathbb{E}[\|\boldsymbol{v}_m\|_2^2]}{\mathbb{E}[\|\boldsymbol{w}_m\|_2^2]} = (\frac{\gamma_{\boldsymbol{v}}^{(m)}}{\gamma_{\boldsymbol{w}}^{(m)}})^2$:

$$C_m \triangleq 1 - \frac{2\alpha_{\max}\sqrt{\frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}}}}{\left(1 + \frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}}\right)} = 1 - \frac{2\alpha_{\max}\sqrt{u_m}}{1 + u_m} \geq 1 - \alpha_{\max}. \tag{10}$$

Notice that $\alpha_{\max}$ does not depend on u_m. Since $0 \leq \alpha_m \leq 1$, the local minima of C_m w.r.t. u_m is equal to $1 - \alpha_{\max}$. Plugging (10) and (9) into (8) results in:

$$\begin{aligned}
\psi &\leq (1 - \alpha_{\max})^{-M} \prod_{m=1}^{M} \frac{P_{\boldsymbol{s}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}\left(1 + \frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}}\right)}, \\
&= \frac{(1 - \alpha_{\max})^{-M}\,\phi}{\left(1 + r + \frac{\phi}{\eta}\right)} \leq \frac{(1 - \alpha_{\max})^{-M}\,\phi}{1 + \frac{\phi}{\eta}},
\end{aligned} \tag{11}$$

where $\phi = \prod_{m=1}^{M} \frac{P_{\boldsymbol{s}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}}$ and $\eta = \prod_{m=1}^{M} \frac{P_{\boldsymbol{s}}^{(m)}}{P_{\boldsymbol{v}}^{(m)}}$ are the SDR and SQNR to the M power in natural units, respectively, and $r \triangleq \prod_{m=1}^{M}(1 + \frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}}) - 1 - \frac{\phi}{\eta} \geq 0$ is the residual part. Pulling (11) back into (3) results in (4). ■

Algorithm 1: *Mix2*(s_1, s_2, β).

$\alpha \leftarrow 10\log_{10} \frac{\|s_1\|_2^2}{\|s_2\|_2^2}$
if $\beta > \alpha$ **then**
  $s_2 \leftarrow \sqrt{\frac{\alpha}{\beta}} \cdot s_2$ ▷ Decrease s_2
else
  $s_1 \leftarrow \sqrt{\frac{\beta}{\alpha}} \cdot s_1$ ▷ Decrease s_1
end if
return $s_1 + s_2$

B. PROOF: ASYMPTOTIC QUANTIZED SDR

Proof: In the asymptotic cases, $P_{\boldsymbol{w}}^{(m)} \gg P_{\boldsymbol{v}}^{(m)}$ and $P_{\boldsymbol{w}}^{(m)} \ll P_{\boldsymbol{v}}^{(m)}$ are equivalent to $\frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}} \to 0$ and $\frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}} \to \infty$, respectively. Then, from (8):

$$\begin{aligned}
\psi &= \prod_{m=1}^{M} \frac{P_{\boldsymbol{s}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}\left(\frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}} + 1 + 2\sqrt{\frac{P_{\boldsymbol{v}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}}}\mathbb{E}\left[\bar{\boldsymbol{w}}_m^T \bar{\boldsymbol{v}}_m\right]\right)} \\
&= \begin{cases} \prod_{m=1}^{M} \frac{P_{\boldsymbol{s}}^{(m)}}{P_{\boldsymbol{w}}^{(m)}} = \phi & P_{\boldsymbol{w}}^{(m)} \gg P_{\boldsymbol{v}}^{(m)} \quad \forall m, \\ \prod_{m=1}^{M} \frac{P_{\boldsymbol{s}}^{(m)}}{P_{\boldsymbol{v}}^{(m)}} = \eta & P_{\boldsymbol{w}}^{(m)} \ll P_{\boldsymbol{v}}^{(m)} \quad \forall m. \end{cases}
\end{aligned} \tag{12}$$

Pulling it back into (3) results in (5). ■

C. SNR SIMULATION

To simulate a wide range of SNRs, we use Algorithm 1, which mixes two signals, $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, with the desired SNR, β. Then, we scale the mixed signal if needed, as in [30].

REFERENCES

[1] D. Michelsanti et al., "An overview of deep-learning-based audio-visual speech enhancement and separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1368–1396, 2021.

[2] 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.

[3] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," in *Proc. Interspeech*, 2020, pp. 2642–2646.

[4] 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.

[5] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, vol. 23, pp. 1–5.

[6] S. K. Gaikwad, B. W. Gawali, and P. Yannawar, "A review on speech recognition technique," *Int. J. Comput. Appl.*, vol. 10, no. 3, pp. 16–24, 2010.

[7] A. Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[8] R. Valle, K. J. Shih, R. Prenger, and B. Catanzaro, "Flowtron: An autoregressive flow-based generative network for text-to-speech synthesis," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=Ig53hpHxS4

[9] Y. Ren et al., "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=piLPYqxtWuA

[10] K. Tan and D. Wang, "Towards model compression for deep learning based speech enhancement," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1785–1794, 2021.

[11] F. Wen, L. Chu, P. Liu, and R. C. Qiu, "A survey on nonconvex regularization-based sparse and low-rank recovery in signal processing, statistics, and machine learning," *IEEE Access*, vol. 6, pp. 69883–69906, 2018.

[12] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," in *Low-Power Computer Vision*. NY, USA: Chapman Hall/CRC, 2022, pp. 291–326.

[13] H. V. Habi, R. H. Jennings, and A. Netzer, "HMQ: Hardware friendly mixed precision quantization block for CNNs," in *Proc Eur. Conf. Comput. Vis.*, 2020, pp. 448–463.

[14] H. V. Habi et al., "HPTQ: Hardware-friendly post training quantization," 2021, *arXiv:2109.09113*.

[15] B. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2704–2713.

[16] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=rkgO66VKDS

[17] M. Nagel, R. A. Amjad, M. Van Baalen, C. Louizos, and T. Blankevoort, "Up or down? adaptive rounding for post-training quantization," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7197–7206.

[18] J. Xu, J. Yu, X. Liu, and H. Meng, "Mixed precision DNN quantization for overlapped speech separation and recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 7297–7301.

[19] Y. Wu, C. Li, and Y. Qian, "Light-weight visualvoice: Neural network quantization on audio visual speech separation," in *Proc. 2023 IEEE Int. Conf. Acoust., Speech, Signal Process. Workshops*, 2023, pp. 1–5.

[20] A. Prasad, P. Motlicek, and S. Madikeri, "Quantization of acoustic model parameters in automatic speech recognition framework," 2020, *arXiv:2006.09054*.

[21] X. Li et al., "Q-diffusion: Quantizing diffusion models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 17535–17545.

[22] S. Kim et al., "Integer-only zero-shot quantization for efficient speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 4288–4292.

[23] E. Cohen, H. V. Habi, and A. Netzer, "Towards fully quantized neural networks for speech enhancement," in *Proc. Interspeech 2023*, 2023, pp. 181–185.

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

[25] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, pp. 1789–1819, 2021.

[26] X. Hao, X. Su, Z. Wang, Q. Zhang, H. Xu, and G. Gao, "SNR-based teachers-student technique for speech enhancement," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2020, pp. 1–6.

[27] S. Chen et al., "Ultra fast speech separation model with teacher student learning," in *Proc. Interspeech*, 2021, pp. 3026–3030.

[28] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*.

[29] S. Lutati, E. Nachmani, and L. Wolf, "Sepit: Approaching a single channel speech separation bound," in *Proc. Interspeech*, 2022, pp. 5323–5327.

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

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations 2015 San Diego, CA, USA, May 7-9, 2015, Conf. Track Proc.*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[32] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 5206–5210.

[33] G. Wichern et al., "WHAM!: Extending speech separation to noisy environments," in *Proc. Interspeech*, 2019, pp. 1368–1372.

[34] M. Pariente et al., "Asteroid: The PyTorch-based audio source separation toolkit for researchers," in *Proc. Interspeech*, 2020, pp. 2637–2641.

[35] 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.

[36] N. S. Detlefsen et al., "TorchMetrics - measuring reproducibility in pytorch," *J. Open Source Softw.*, vol. 7, no. 70, 2022, Art. no. 4101, doi: 10.21105/joss.04101.

[37] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "MUSDB18-HQ - an uncompressed version of MUSDB18," Aug. 2019, doi: 10.5281/zenodo.3338373.

[38] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017, doi: 10.5281/zenodo.1117372.

[39] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," 2020. [Online]. Available: https://openreview.net/forum?id=HJx7uJStPH

[40] F.-R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *Proc. Latent Variable Anal. Signal Separation: 14th Int. Conf.*, 2018, pp. 293–305.