INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Towards Fully Quantized Neural Networks For Speech Enhancement

Elad Cohen, Hai Victor Habi, Arnon Netzer

Sony Semiconductor Israel, Israel
{elad.cohen02,hai.habi,arnon.netzer}@sony.com

Abstract

Deep learning models have shown state-of-the-art results in speech enhancement. However, deploying such models on an eight-bit integer-only device is challenging. In this work, we analyze the gaps in deploying a vanilla quantization-aware training method for speech enhancement, revealing two significant observations. First, quantization mainly affects signals with a high input Signal-to-Noise Ratio (SNR). Second, quantizing the model's input and output shows major performance degradation. Based on our analysis, we propose Fully Quantized Speech Enhancement (FQSE), a new quantization-aware training method that closes these gaps and enables eight-bit integer-only quantization. FQSE introduces data augmentation to mitigate the quantization effect on high SNR. Additionally, we add an input splitter and a residual quantization block to the model to overcome the error of the input-output quantization. We show that FQSE closes the performance gaps induced by eight-bit quantization.

Index Terms: Speech Enhancement, Quantization, CNN

1. Introduction

Deep learning models have shown state-of-the-art results on many audio tasks such as source separation [1, 2, 3, 4, 5], speech enhancement [6] and speech recognition [7]. Nevertheless, deploying such models on efficient edge or mobile devices is challenging due to memory and computational complexity limitations. One direction to address these problems is quantization [8] which reduces models' memory and computational complexity. Due to the success of quantization in Computer Vision [9, 10], there are attempts to apply quantization to several audio tasks, such as speech recognition [11] and enhancement [12]. However, providing a fully quantized model is still challenging when both activations and weights have low precision (i.e. eight-bit or below). A recent work [13] in this direction uses quantization with a mixed precision that includes high precision bit-width (i.e. above eight-bit). Another method [14] uses clustering quantization which only reduces the model size. Still, efficient deployment of such models on embedded and mobile devices remains challenging.

In this work, we suggest a Quantization-Aware Training (QAT) [8] method for speech enhancement models that enables a fully eight-bit quantized model, where the weights and activations of every operation are quantized. To achieve this, we first analyze the performance of a vanilla eight-bit QAT [15]. This analysis shows that speech enhancement quantized models are highly sensitive to input SNR. In addition, it reveals a high sensitivity to activation quantization, especially for the input and output signals. Based on this analysis, we introduce a new QAT method for speech enhancement models. We call our method Fully Quantized Speech Enhancement (FQSE). FQSE is based on LSQ [9], which enables learning the quantizer step size. In addition, we combine FQSE with a new data augmentation method that considers high SNR levels to reduce the quantization error in these SNRs. Finally, to cope with the sensitivities of the input and output quantization, we introduce two modifications to the trained model: i) splitting a single speech channel with high precision into a pair of low-precision channels; ii) adding a residual quantization block, which produces an additional output to the model that extracts the output quantization error. The suggested modifications require adding pre and post processing stages with a small computational cost. We present several experiments on Conv-TasNet [1] model to show the effectiveness of FQSE on speech enhancement. Moreover, we validate our approach across several input SNRs to present its performance when facing high SNR levels. Our contributions are summarized as follows:

- We analyze a fully quantized speech enhancement neural network and show its sensitivity to high input SNR and input-output quantization.
- We introduce a Refined Quantization-aware training Strategy (RQS) that enables a fully eight-bit quantization of speech enhancement models.
- We suggest an Input Splitter and a Residual Quantization Block (RQB), which are added to a pre-trained model and enable input-output quantization.

For reproducible research, we share the code here [16].

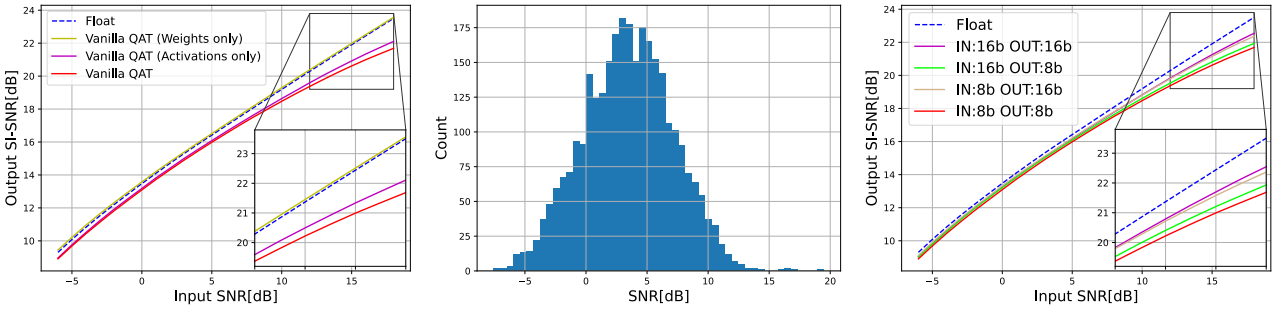
2. Background

Here, we provide a short overview of QAT [8, 9, 15]. QAT is a crucial method for enabling the deployment of neural networks on embedded devices. Most QAT methods begin with a pre-trained neural network and add weights and activations quantizers. Then, retraining the network to correct the quantization-induced errors. In this work, we use uniform quantization with *asymmetric* and *symmetric* thresholds for activation and weights, respectively. A uniform quantizer is defined as follows: let $\boldsymbol{x} \in \mathbb{R}^n$ be a vector to be quantized, $\Delta \in \mathbb{R}^+$ is the quantizer step-size, $z \in \mathbb{R}$ is the zero-point and b is the bit-width, then the uniform quantization function is defined as:

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

where $\mathrm{clip}\left(\boldsymbol{x}, a, b\right) \triangleq \min\left(\max\left(\boldsymbol{x}, a\right), b\right)$ denotes the clipping of $\boldsymbol{x}$ between a, b, and $\lfloor x \rceil : \mathbb{R} \rightarrow \mathbb{Z}$ denotes the rounding of x to the nearest integer value. In case of asymmetric thresholds, we set $\Delta = \frac{\max(\boldsymbol{x})-\min(\boldsymbol{x})}{2^b-1}$ and $z = \min\left(\boldsymbol{x}\right)$ whereas in (signed) symmetric quantizer $z = -t$ and $\Delta = \frac{t}{2^{b-1}}$, where

10.21437/Interspeech.2023-883



(a) *Float vs Quantized* (b) *SNR Histogram* (c) *In-Out Precision*

Figure 1: *Analysis of a fully eight-bit quantized Conv-TasNet [1] model. (a) presents the* SI-SNR *of the float model vs. eight-bit vanilla quantized model for different input SNR levels. (b) SNR histogram of the LibriMix [17] test set. (c) shows the effect of input-output precision on the* SI-SNR.

$t = \max(|\boldsymbol{x}|)$ is the threshold value. Retraining a quantized neural network is challenging due to the rounding operation in (1), which is non-differentiable. There are several approaches in the literature to overcome this problem. Here we use one of the simplest approaches, a Straight-Through Estimator (STE) [18], which approximates the gradient of the rounding operation as one, namely $\frac{\partial \lfloor y \rceil}{\partial y} = 1$.

3. Analysis of Quantization Error

We analyze the effect of QAT on the speech enhancement task. In this analysis, we perform a vanilla QAT [15] on the Conv-TasNet [1] model while using the LibriMix [17] dataset. Unless stated otherwise, the model is fully quantized with eight-bit quantization for both weights and activations. The implementation details are described in Section 5.1.

In real-world scenarios, speech quality may vary. It can be much noisier or cleaner. One would expect the enhancement model to produce clean speech, especially when the inputs are cleaner (high SNR). We believe that quantization noise becomes dominant and its effect is much stronger in such cases. To illustrate the effect of different input SNR levels, we artificially scale the input noises in the entire test set of LibriMix to get the desired SNR. Then, we compute the average Scale-Invariant Signal-to-Noise Ratio (SI-SNR) [19] over the entire noise-scaled test set. Assume that for each sample in the dataset, we have the clean speech signal $\boldsymbol{x}$ and its noisy version $\tilde{\boldsymbol{x}}$. Using both signals, we generate a noisy measurement $\tilde{\boldsymbol{y}}$ as follows:

$$\tilde{\boldsymbol{y}} = \boldsymbol{x} + \alpha \boldsymbol{n}, \tag{2}$$

where $\alpha = \sqrt{10^{-\frac{\beta}{10}} \cdot \frac{\|\boldsymbol{x}\|_2^2}{\|\boldsymbol{n}\|_2^2}}$ is the noise scale factor, $\boldsymbol{n} = \tilde{\boldsymbol{x}} - \boldsymbol{x}$ is the orignal noise and β is the desired SNR in Decibel. We repeat this procedure with multiple SNR levels (β) between $-6dB$ to $18dB$ at $1dB$ step and report the results in Figure 1a. Note that this range has been selected to cover input SNR variations based on the histogram (Figure 1b) of the original LibriMix test set. Figure 1a shows that the vanilla QAT performance degrades in high input SNRs, resulting in a wider gap between the float and the quantized model, whereas in low input SNRs, the gap is smaller. This emphasizes the model's sensitivity to quantization in high input SNRs.

Next, we investigate what causes the increased sensitivity to high input SNRs. First, we run the same experiment but only quantize the activations or weights. This experiment (Figure 1a) shows that the performance degradation results from quantizing the activations. To investigate this further, we perform an additional experiment where both weights and activations are quantized, but the input and output tensors are kept at high precision (16-bit). Figure 1c shows significant sensitivity to input and output quantization and that both quantizations contribute to performance degradation.

4. Method

We suggest Fully Quantized Speech Enhancement (FQSE), a method to obtain a fully quantized neural network for speech enhancement, in which all model weights and activations are quantized with low precision eight-bit. Based on our analysis in Section 3, we derive a QAT method that consists of three parts: 1) a Refined QAT Strategy (RQS), which applies data augmentation to correct the imbalanced dataset and takes into account the quantization sensitivity to high SNRs; 2) an input correction step that splits a single high-precision channel into a pair of low-precision channels; 3) a Residual Quantization Block (RQB) which outputs the residual quantization error that is later combined into a high-precision output. Our approach is illustrated in Figure 2. We describe RQS in Section 4.1 and the modification to the input and output signals in Sections 4.2 and 4.3, respectively.

4.1. Refined QAT Strategy

Here, we suggest a Refined QAT Strategy (RQS) based on learned step size quantization (LSQ) [9]. Adding LSQ to vanilla QAT allows us to consider activation quantization error by learning the quantizer step size using gradients of the task loss. We address the imbalanced SNR distribution of the training set (Figure 1b) and put more attention on high SNR samples during training. This is achieved by an SNR augmentation method. We randomly select an SNR value for each sample at every batch and then rescale the sample noise as described in Equation (2). During the retraining process, we sample the SNR uniformly between $-6dB$ to $18dB$.

4.2. Input Splitter

A quantization of input speech into low precision degrades the performance of speech enhancement models (Figure 1c). In order to keep high precision in the input, we use two signals of eight-bit instead of one signal of 16-bit. This is achieved

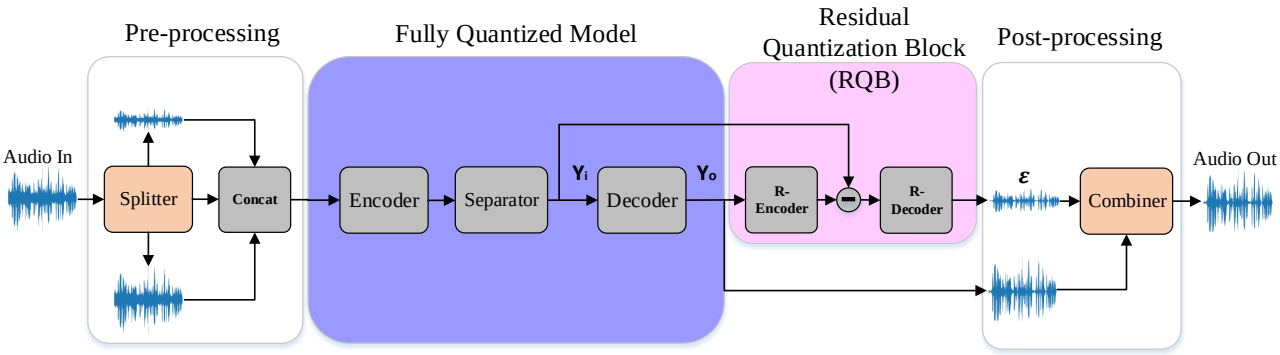


Figure 2: *FQSE scheme. The pre-processing stage is an Input Audio Splitter (SPT) that splits a high-precision signal into two signals. The model, which consists of Encoder-Decoder architecture with the new Residual Quantization Block (RQB) is then fully quantized to low precision. The post-processing stage combines two low-precision outputs into a single high-precision output.*

by splitting the input in the pre-processing stage from a single channel into a pair of channels using a simple bit-splitting technique.

The splitting is done in an orderly fashion from the most significant bit to the least significant bit, where each split takes eight consecutive bits. We implement this bit-splitting with a symmetric floor quantizer. Specifically, the low-precision speech is given by:

$$\mathbf{X} = \begin{bmatrix} \mathrm{Q_F}\left(\boldsymbol{x}, b_l, t\right) \\ \mathrm{Q_F}\left(\frac{2 \cdot t \cdot \boldsymbol{\epsilon}}{\Delta(b_l,t)} - t, b_l, t\right) \end{bmatrix}, \tag{3}$$

where $\mathrm{Q_F}$ is a symmetric floor quantizer. In Equation (3) the residual error is $\boldsymbol{\epsilon} \triangleq \boldsymbol{x} - \mathrm{Q_F}\left(\boldsymbol{x}, b_l, t\right)$ and the symmetric floor quantizer is given by:

$$\mathrm{Q_F}\left(x, b_l, t\right) \triangleq \Delta\left(b_l, t\right) \cdot \mathrm{clip}\left(\left\lfloor \frac{x}{\Delta\left(b_l, t\right)} \right\rfloor, -2^{b_l - 1}, 2^{b_l - 1} - 1\right),$$

where $\lfloor x \rfloor : \mathbb{R} \rightarrow \mathbb{Z}$ denotes the floor of x to the lowest integer value. In order to restrict the residual error in the same range $[-t, t)$ as $\mathrm{Q_F}\left(\boldsymbol{x}, b_l, t\right)$, we scale it by $\frac{2 \cdot t}{\Delta(b_l,t)}$ and subtract t. This way, $\mathbf{X}$ remains with per tensor quantization. In this work, we denote the bit-width of a low precision tensor as $b_l = 8$ and high precision tensor as $b_h = 16$. This split requires changing the first linear operation (Figure 2, Encoder) weights to two channels (instead of one channel) same as the input. We initialize the new weights as follows: the first channel consists of the original weights, whereas the second channel is sampled from Gaussian distribution with mean and variance equal to the original weights.

4.3. Residual Quantization Block

Based on the analysis in Section 3, we have shown that low precision output (eight-bit) degrades the performance of a speech enhancement model (Figure 1c). We address this issue by proposing a Residual Quantization Block (RQB) as shown in Figure 2. The RQB extracts the quantization residual error by using the encoder-decoder structure which is common in many speech enhancement networks [5]. We use the feature space to compute the quantization residual error in low precision, which is feasible thanks to the increased number of channels. Specifically, let $\mathbf{Y}_i \in \mathbb{R}^{d \times n}$, $\mathbf{Y}_o \in \mathbb{R}^{1 \times n}$ be the decoder's quantized input and output tensors, respectively, where d is the number of channels and n is the number of time domain samples. Then, the RQB output is given by:

$$\tilde{\mathbf{Y}}_i = Q\left(\mathbf{W}_{\tilde{\mathbf{E}}} \mathbf{Y}_o\right), \tag{4a}$$

$$\mathbf{U} = Q\left(\mathbf{Y}_i - \tilde{\mathbf{Y}}_i\right), \tag{4b}$$

$$\boldsymbol{\epsilon} = Q\left(\mathbf{W}_{\tilde{\mathbf{D}}} \mathbf{U}\right), \tag{4c}$$

where $\mathbf{W}_{\tilde{\mathbf{E}}} \in \mathbb{R}^{d \times 1}$ and $\mathbf{W}_{\tilde{\mathbf{D}}} \in \mathbb{R}^{1 \times d}$ are the quantized weights of RQB's encoder and decoder, respectively. The RQB first projects the output of the quantized model into the feature space in Equation (4a). Then, RQB computes the quantization residual error in Equation (4b). Finally, the error is projected back into the time domain in Equation (4c). Note that RQB is also fully quantized as shown in Equation (4). We initialize the RQB encoder-decoder block using the parameters of the pre-trained model and then learn them during the QAT optimization process. Finally, in the post-processing stage, we reconstruct the high-precision output by $\boldsymbol{y} = Q\left(\mathbf{Y}_o\right) + \frac{\boldsymbol{\epsilon}}{2^{b_l - 1}}$.

5. Experimental Results

5.1. Implementation Details

Training. We use the LibriMix [17] dataset for retraining and testing our method. The LibriMix dataset is derived from LibriSpeech [20] signals (clean subset), and WHAM [21] noises. It consists of three splits: train, dev, and test. Each split contains short mixed/noisy audios of 16-bit samples. For training, we use the train split (train-360), which contains 50,800 samples. In this work, we use the LibriMix dataset with a sample rate of 16kHz and the shortest waveform length between the noisy and clean signals. We evaluated FQSE on Conv-TasNet [1], which is a convolution-based neural network (CNN). We begin with pre-trained float weights taken from [22]. We quantize the model for eight-bit integer-only, with per-tensor and per-channel thresholds for activations and weights, respectively. The optimization process minimizes negative SI-SNR [19] for 80 epochs using Adam [23] optimizer with a learning rate of 10^{-3}. Each epoch takes approximately 45 minutes on an NVIDIA DGX station with four NVIDIA V100 32GB GPUs and a batch size of 6.

Evaluation. We use the LibriMix test split, which contains 3000 samples. For evaluating our method we use the following

Table 1: *Comparing between float and eight-bit quantized models on LibriMix [17] test set using Conv-TasNet [1]. For the* SI-SNR *metric, there are also results for SNR per range. GigaBit Operations (GBOP) is for a 3-second segment.*

Precision	Model Size [MB]	GBOP	SI-SNR[dB]				SDR [dB]	STOI
			Low SNR	Mid SNR	High SNR	All SNR		
Float	20.1	329.12	12.50	15.72	19.08	14.74	15.30	0.9311
Vanilla QAT	5.15	20.57	12.26	15.38	18.45	14.42	14.94	0.9253
FQSE(Ours)	5.20	23.72	12.57	15.74	19.02	14.77	15.26	0.9294

Table 2: *Comparing Splitter (SPT) and Combiner (RQB) to low precision using LibriMix [17] test set.*

Input Precision	Output Precision	SI-SNR[dB]		
		Low SNR	Mid SNR	High SNR
8bit	16bit	12.46	15.69	18.99
SPT	16bit	12.56	15.78	19.14
16bit	8bit	12.46	15.66	18.93
16bit	RQB	12.47	15.67	19.38

metrics[1]: SI-SNR, Signal-to-Distortion Ratio (SDR) [19] and Short-Time Objective Intelligibility measure (STOI) [25]. We present results on different input SNR levels to show that the suggested approach reduces the sensitivity of eight-bit quantization. This is achieved by splitting the LibriMix test set into three ranges: low, mid, and high, which are below 2dB, between 2dB to 10dB, and above 10dB, respectively. Also, we add results using the original LibriMix test set.

5.2. Results

We begin with an ablation study to show the benefit of our method. First, we present the improvement of each component of FQSE on several SNR levels as in Section 3. Then, in Figure 3 we present the results of vanilla QAT, float, FQSE, FQSE (w/o RQS), and FQSE (w/o SPT and RQB). We observe that all FQSE components are required to reach the float model performance. In addition, it shows an improvement across a wide range of SNR levels and in particular high SNRs where we reach an improvement of 2dB compared to vanilla QAT. We also conduct experiments to quantify the benefit of the splitter and RQB. Table 2 shows the effectiveness of both the splitter and RQB for keeping high precision in the input and output, respectively.

Finally, in Table 1 we present the results of quantized Conv-TasNet [1] on the entire LibriMix [17] test set. We show that FQSE, which is fully quantized, has a similar performance to the float model while reducing the model size by a factor of four and the computation complexity is $\sim$15% bigger than vanilla QAT.

6. Conclusions

In this work, we present a QAT method for speech enhancement models with eight-bit integer-only efficient inference called FQSE. Our performance analysis shows sensitivities to inputs with high SNR as well as quantization of the model's input and output. We address these sensitivities by suggesting an SNR data augmentation and adjustment to the input and output of the quantized model. We present an ablation study showing the contribution of each component of FQSE and the results on an eight-bit integer-only model. This is the first step towards eight-bit integer-only quantization, and several questions still remain open: i) can this method be generalized to other tasks and architectures?; ii) how to obtain an integer-only post-training quantization method?

[1]We use TorchMetrics [24] for the metrics implementation.

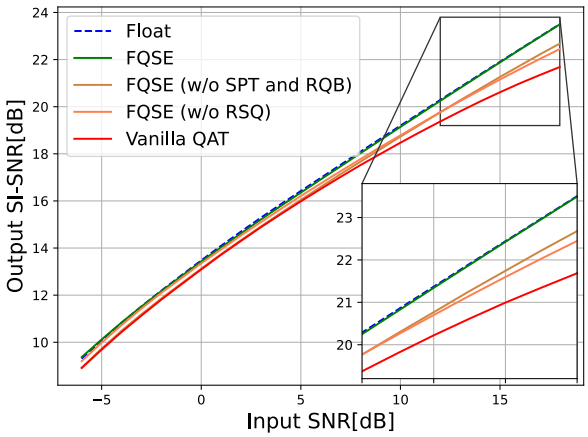


Figure 3: SI-SNR *comparison between vanilla QAT and FQSE with eight-bit quantization for different input SNRs. FQSE reaches the float model across all input SNRs. FQSE w/o RSQ and FQSE w/o the splitter (SPT) and RQB show the need for both techniques.*

7. Acknowledgement

We would like to thank Stefan Uhlich for his helpful discussions and suggestions.

8. References

[1] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[2] 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*, 2020, pp. 2642–2646.

[3] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 46–50.

[4] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 21–25.

[5] D. Michelsanti, Z.-H. Tan, S.-X. Zhang, Y. Xu, M. Yu, D. Yu, and J. Jensen, "An overview of deep-learning-based audio-visual speech enhancement and separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1368–1396, 2021.

[6] S.-W. Fu, C. Yu, K.-H. Hung, M. Ravanelli, and Y. Tsao, "Metricgan-u: Unsupervised speech enhancement/dereverberation based only on noisy/reverberated speech," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7412–7416.

[7] S. K. Gaikwad, B. W. Gawali, and P. Yannawar, "A review on speech recognition technique," *International Journal of Computer Applications*, vol. 10, no. 3, pp. 16–24, 2010.

[8] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *arXiv preprint arXiv:2103.13630*, 2021.

[9] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=rkgO66VKDS

[10] H. V. Habi, R. H. Jennings, and A. Netzer, "Hmq: Hardware friendly mixed precision quantization block for cnns," in *European Conference on Computer Vision*. Springer, 2020, pp. 448–463.

[11] H. D. Nguyen, A. Alexandridis, and A. Mouchtaris, "Quantization aware training with absolute-cosine regularization for automatic speech recognition." in *Interspeech*, 2020, pp. 3366–3370.

[12] S. Abdullah, M. Zamani, and A. Demosthenous, "Towards more efficient dnn-based speech enhancement using quantized correlation mask," *IEEE Access*, vol. 9, pp. 24 350–24 362, 2021.

[13] J. Xu, J. Yu, X. Liu, and H. Meng, "Mixed precision dnn quantization for overlapped speech separation and recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7297–7301.

[14] K. Tan and D. Wang, "Towards model compression for deep learning based speech enhancement," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 29, pp. 1785–1794, 2021.

[15] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2704–2713.

[16] E. Cohen, H. V. Habi, and A. Netzer, "Towards fully quantized neural networks for speech enhancement," https://github.com/ssi-research/FQSE, 2023.

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

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

[19] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr–half-baked or well done?" in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[21] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "WHAM!: Extending Speech Separation to Noisy Environments," in *Proc. Interspeech 2019*, 2019, pp. 1368–1372.

[22] M. Pariente, S. Cornell, J. Cosentino, S. Sivasankaran, E. Tzinis, J. Heitkaemper, M. Olvera, F.-R. Stöter, M. Hu, J. M. Martín-Doñas, D. Ditter, A. Frank, A. Deleforge, and E. Vincent, "Asteroid: the PyTorch-based audio source separation toolkit for researchers," in *Proc. Interspeech*, 2020.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[24] N. S. Detlefsen, J. Borovec, J. Schock, A. H. Jha, T. Koker, L. D. Liello, D. Stancl, C. Quan, M. Grechkin, and W. Falcon, "Torchmetrics - measuring reproducibility in pytorch," *Journal of Open Source Software*, vol. 7, no. 70, p. 4101, 2022. [Online]. Available: https://doi.org/10.21105/joss.04101

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