Interspeech 2024
1-5 September 2024, Kos, Greece

# Resource-Efficient Speech Quality Prediction through Quantization Aware Training and Binary Activation Maps

*Mattias Nilsson*[1,*], *Riccardo Miccini*[2,3,*], *Clément Laroche*[2], *Tobias Piechowiak*[2], *Friedemann Zenke*[1,4]

[1]Friedrich Miescher Institute for Biomedical Research, Switzerland, [2]GN Audio, Denmark,
[3]Technical University of Denmark, Denmark, [4]University of Basel, Switzerland, [*]equal contributions

mattias.nilsson@fmi.ch, rimi@dtu.dk, claroche@jabra.com, topiechowiak@jabra.com, friedemann.zenke@fmi.ch

## Abstract

As speech processing systems in mobile and edge devices become more commonplace, the demand for unintrusive speech quality monitoring increases. Deep learning methods provide high-quality estimates of objective and subjective speech quality metrics. However, their significant computational requirements are often prohibitive on resource-constrained devices. To address this issue, we investigated binary activation maps (BAMs) for speech quality prediction on a convolutional architecture based on DNSMOS. We show that the binary activation model with quantization aware training matches the predictive performance of the baseline model. It further allows using other compression techniques. Combined with 8-bit weight quantization, our approach results in a 25-fold memory reduction during inference, while replacing almost all dot products with summations. Our findings show a path toward substantial resource savings by supporting mixed-precision binary multiplication in hard- and software.

**Index Terms**: speech quality prediction, quantization aware training, binary activation maps

## 1. Introduction

Power-constrained audio devices such as headsets, earbuds, and hearing aids are driving the need for resource-efficient and real-time speech enhancement solutions. Denoising based on deep learning—so-called deep noise suppression (DNS)—is surpassing conventional speech enhancement methods based on signal processing by its ability to handle complex real-world noise [1–3]. When evaluating DNS systems, commonly used metrics include objective ones, such as perceptual evaluation of speech quality (PESQ) [4], short-time objective intelligibility (STOI) [5], and scale-invariant signal-to-distortion ratio (SI-SDR) [6], as well as subjective mean opinion scores (MOSs). Each of these metrics features both strengths and weaknesses. Specifically, objective metrics are computationally efficient but do not always correlate well with human perception [7] and require a clean reference signal, which makes them difficult to use with real-world data. Conversely, subjective metrics are more accurate and reliable, but require human listeners and are therefore more expensive and time-consuming to obtain.

In recent years, speech quality prediction (SQP) has emerged as a promising way to overcome the limitations of existing quality metrics by using machine learning. SQP systems can be trained to predict both objective and subjective metrics directly from a noisy or processed speech signal, without the need for clean reference signals or human listeners, thereby making such systems "unintrusive", i.e. reference-less. Notable recent works on unintrusively estimating objective metrics target PESQ [8–10] and STOI [11], as well as multiple metrics simultaneously [12–14]. Similarly, SQP systems aimed at estimating user MOS have also been developed [15–17].

To make real-time DNS more efficient, common approaches include architectural adjustments such as the use of depth-wise separable convolutions [1], model compression techniques such as quantization [3], as well as aspects of dynamic neural networks such as slimmable channels [18], layer skipping [2], or early exiting [19–22]. This latter class of models can adapt its computational graph at inference time depending on the input, which is particularly useful in real-world applications with substantially changing noise conditions. To accomplish this, dynamic neural networks rely on exiting or skipping policies that are often hand-crafted using simple heuristics [20, 21], or small sub-networks that are trained end-to-end with the DNS system [2], often using differentiable relaxation of a discrete distribution [23] which is difficult to train. Importantly, none of the above strategies take speech quality into account.

While some attempts have been made at estimating concrete characteristics of the speech signal with the purpose of affecting the routing of a downstream model [22], this area of research is little explored—arguably due to the computational demands currently associated with even the smallest SQP model. Here, we argue that an adequately efficient and lean SQP system will benefit dynamic noise suppression, allowing for continuous inference in real time. Furthermore, the aforementioned ubiquity of DNS systems, especially on embedded and wearable devices, which might experience significantly different conditions than originally trained for, calls for tighter scrutiny and monitoring of performance to avoid further signal degradation in the form of unwanted artifacts [24].

Current SQP solutions range from having tens of thousands of trainable parameters [15] to several millions [12]. However, due to the long sequences used for prediction, storing the activations of the early layers remains demanding, why reducing their memory footprint through quantization is a primary goal. While post-training quantization (PTQ) schemes exist, they perform poorly for binarization—an extreme form of quantization. Still, binarization is particularly appealing in the SQP setting because it minimizes memory consumption while also allowing for replacement of many algorithmic computations with bitwise operations. To reach good task performance with binarized neural networks, quantization aware training (QAT) is imperative. Standard QAT techniques for binarized neural networks are surrogate gradients, which are nonlinear variants of straight-through estimators [23, 25, 26].

In this work, we investigated reduction of the computational cost of SQP, thereby paving the way for speech-quality driven dynamic denoising. Specifically, we examined the computational benefits of binary activation maps (BAMs) and weight


10.21437/Interspeech.2024-1979

quantization in the DNSMOS architecture [15]—a convolutional neural network (CNN) commonly used for evaluating denoising systems. For binarizing the activations, we used a QAT method based on a straight-through estimator [23] with a non-linear surrogate derivative commonly used for optimizing spiking neural networks [26]. We show that BAMs combined with non-binary weights result in substantial resource savings while hardly affecting the performance of the model in terms of common evaluation metrics.

# 2. Methods

## 2.1. Problem Definition

Given a reference speech signal $s$, we denote its degraded analogue as $s_d$. Conventional intrusive algorithms, such as PESQ [4], compute a speech quality metric SQ using both signals:

$$\mathrm{SQ} = f(s, s_d). \tag{1}$$

In the context of SQP, we aim to train a deep learning model to estimate the metric using only the degraded signal:

$$\arg\min_{\theta \in \mathbb{R}} \mathcal{L}(\mathrm{SQ}, \hat{\mathrm{SQ}}); \quad \hat{\mathrm{SQ}} = f_\theta(s_d), \tag{2}$$

where $\hat{\mathrm{SQ}}$ is the estimated speech quality metric, $\mathcal{L}$ is a cost or loss function, and $\theta$ represents the model parameters.

## 2.2. Quantization

In this work, we compared different quantization approaches. Specifically, we looked at binary activations, binary weights, and 8-bit quantization of weights and activations.

### *2.2.1. Binary Activation Maps (BAMs)*

To binarize the activation maps of DNSMOS, we replaced the activations of the convolutional layers of the model with the Heaviside step function:

$$H(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}. \tag{3}$$

This makes the activations discontinuous and, therefore, non-differentiable. Note that we refer to networks with BAMs as binarized networks, even when their weights are non-binary. In addition to the change in activation function in the convolutional layers, we used global average pooling instead of global max pooling after the final convolutional layer in all binarized networks. In Sec. 2.3, we present a QAT method that we used to circumvent the problem of non-differentiable activations.

### *2.2.2. Binary Weights*

In simulations with binary weights, we wrapped the floating-point weight parameters of the convolutional kernels with a scaled and shifted step function:

$$Q(w) = \begin{cases} -1 & \text{for } w < 0 \\ 1 & \text{for } w \geq 0 \end{cases}. \tag{4}$$

This shift in baseline was done to ensure zero mean preactivations.

### *2.2.3. 8-Bit Quantization*

The 8-bit quantization was done in the form of static PTQ using in-built functionality in PyTorch. This process involves a calibration step in which batches of data are fed through the model and observer modules are used to record the resulting distributions of activations and weights. As input data for the calibration, we used a subset of 20 % of the training data. We used the default PTQ configuration in PyTorch for the "x86" backend. During calibration, activations are divided into 2,048 dynamic histogram bins, which are then subject to a search for the optimal minimum and maximum values that minimize the quantization error. Conversely, weights are quantized by recording their maximum and minimum values on a per-channel basis. These minima and maxima are then employed to derive the scaling and zero-point coefficients of the affine quantization transform [27].

## 2.3. Surrogate Gradient

To circumvent the problem of non-differentiability during QAT of the binarized networks, we used a surrogate gradient approach [26], which extends on the straight-through estimator proposed in [23]. To that end, we replaced the ill-defined derivatives of the discontinuous Heaviside activation function that appear in the backward pass with the derivatives of continuous surrogate activation functions (Fig. 1). Specifically, we used the SuperSpike surrogate derivative [28], which corresponds to the normalized derivative of a fast sigmoid:

$$\tilde{H}'(x) = \frac{1}{(\beta|x| + 1)^2}, \tag{5}$$

where $\beta$ is a steepness parameter.

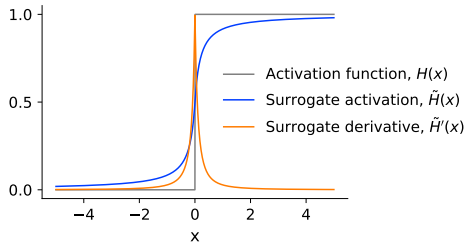


Figure 1: ***Surrogate derivative for training of neural networks with binary activations.*** *The surrogate activation function is a fast sigmoid with adjustable steepness.*

The above surrogate gradient was used as a drop-in replacement for a regular gradient during the backward pass, and, otherwise, training proceeded through standard back-propagation.

# 3. Experimental Setup

We applied and present our findings on the DNSMOS model [15], which is relatively compact and often used to unintrusively estimate MOS on speech enhancement works [29, 30]. The code used to conduct our experiments is available at `https://github.com/fmi-basel/binary-activation-maps-sqp`.

## 3.1. Model

The model consists of a CNN taking in a 2-dimensional representation of an audio signal in the form of a log-power Mel-scale spectrogram. The input is mapped into progressively more

compact representations through convolutional and pooling layers. The resulting features are then averaged along the spectral and temporal dimensions and further processed by a series of fully connected layers to derive a real-valued scalar estimation of a MOS. Compared with the original DNSMOS architecture, we used a wider short-time Fourier transform (STFT) window of 40 ms (while maintaining the 50 % overlap) to ensure correct processing of Mel filter banks. The model and its parameters are presented in Table 1. For an exhaustive description of the architecture and preprocessing steps, see [15].

### 3.2. Dataset

Since the original DNSMOS model was trained on a proprietary, undisclosed dataset, we trained and evaluated our models on the public DNS2020 dataset [32]. This is a synthetic dataset that consists of clean speech with added noise at various signal-to-noise ratios (SNRs). Due to the lack of MOS target labels for this dataset, we computed PESQ using its standard intrusive implementation. Thus, we tasked our model with unintrusively estimating PESQ using only the degraded (i.e., noisy) signal, as stated in the problem formulation (Methods Sec. 2.1). To this end, we used the official DNS2020 generation scripts[1] to derive a dataset consisting of 6,000 noisy–clean pairs of 30 s each, for a total of 50 h of audio. Subsequently, we framed each pair into 9-s segments with a stride of 2 s (corresponding to 10 data points per clip) and computed a PESQ label for each segment, after which we discarded the clean reference signals. In total, this amounts to 150 h of overlapping audio segments. We held out a validation set consisting of 5 % of the original 30-s clips. For testing, we used the synthetic reverberation-free test data that is provided in the DNS2020 dataset, consisting of 150 10-s samples.

### 3.3. Training and Evaluation Metrics

All models were trained using stochastic gradient descent with a batch size of 128 and a learning rate of $10^{-3}$ for 400 epochs using the Adam optimizer with standard parameters and the mean squared error (MSE) loss. We established a scheduling policy to decrease the learning rate by a factor of 0.9 after 5 epochs without improvements, as well as an early stopping policy to automatically interrupt the training after 25 epochs without improvements. The training took about 9.5 hours for the baseline model and 12 hours for the binarized model on an NVIDIA Quadro RTX 5000 GPU. In line with previous work on SQP, we evaluated the performance of the models using MSE and the Pearson correlation coefficient (PCC) [8, 12, 15].

## 4. Results

To increase the computational efficiency of SQP, we sought to investigate the performance impact of BAMs in the convolutional layers of the DNSMOS model. First, we trained and evaluated the full-precision model (Table 1), thereby establishing a baseline for comparison. Given the high number of convolutional operations in DNSMOS, binarizing the activation maps affects virtually all $2.3\times10^6$ activations, and replaces $171\times10^6$ of the $188\times10^6$ multiplications of the model (Table 1) with multiplications in which one of the factors is binary. With suitable soft- and hardware support, such int1 multiplications have the potential to significantly reduce the computational and memory demands of the model. We first measured the impact of post-training binarization of activation maps in the trained baseline model by simply replacing the activations in its convolutional layers with the Heaviside function. This change resulted in a poorly performing model, with a reduction of test PCC from $0.84 \pm 0.03$ to $0.54 \pm 0.02$, see Fig. 2. Thus, binarization of the activation maps constitutes too drastic a precision reduction for performance to be maintained without employing some form of QAT.

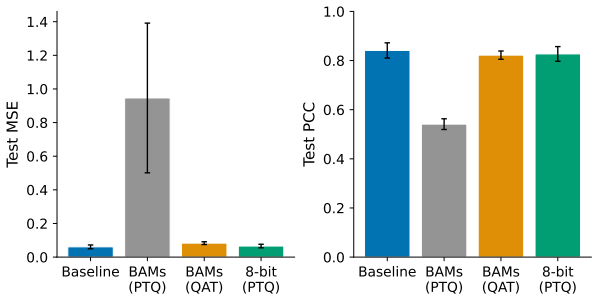


Figure 2: ***Evaluation metrics.*** *Test MSE and PCC for the following different quantizations of DNSMOS: binary PTQ of activation maps (gray), BAMs with QAT (orange), and 8-bit PTQ of activations and weights (green). The black lines indicate standard deviations from repeated instances of training (n=4).*

To investigate the performance benefit of QAT on the binarized model, we conducted the binarization as before and trained the model from scratch using surrogate gradients (Methods Sec. 2.3). We first performed a coarse grid search to determine the optimal value of the steepness parameter, $\beta$. We determined the optimal value $\beta = 5$ on held-out data and used this value for all following experiments. In contrast to the post-training binarization above, we found that QAT rescued the performance of the model, resulting in performance levels that were not significantly different from the baseline model, with $\mathrm{PCC}_{\mathrm{BAM-QAT}} = 0.82 \pm 0.02$ and $\mathrm{PCC}_{\mathrm{base}} = 0.84 \pm 0.03$, respectively, on the test set (Figs. 2 and 3).

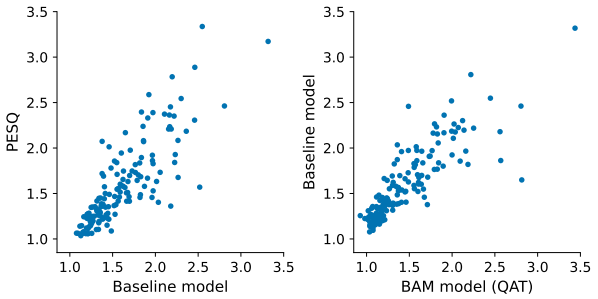


Figure 3: ***Baseline model predictions vs. PESQ and BAM vs. baseline model predictions on the test set.***

Next, we tried binarizing the weights in addition to the activations, using QAT for training the model with fully binarized convolutions from scratch (Methods Sec. 2.2). This change, however, resulted in substantially worse performance with a test PCC of $0.63 \pm 0.05$. We wondered whether there was some middle ground. To test this idea, we next combined 8-bit PTQ of

[1] https://github.com/microsoft/DNS-Challenge/blob/interspeech2020/master/noisyspeech_synthesizer_singleprocess.py

Table 1: ***Baseline model for speech quality prediction, DNSMOS.*** *The numbers of parameters and multiply–add operations were retrieved using torchinfo [31].*

| Layer | Output shape | Parameters | Multiply–adds | Activations |
|---|---|---|---|---|
| Input | 449x120x1 | | | |
| Conv2D-1 (3x3), ReLU | 449x120x32 | 320 | 17,241,600 | 1,724,160 |
| MaxPool2D-1 (2x2), Dropout(0.3) | 224x60x32 | | | |
| Conv2D-2 (3x3), ReLU | 224x60x32 | 9,248 | 124,293,120 | 430,080 |
| MaxPool2D-2 (2x2), Dropout(0.3) | 112x30x32 | | | |
| Conv2D-3 (3x3), ReLU | 112x30x32 | 9,248 | 31,073,280 | 107,520 |
| MaxPool2D-3 (2x2), Dropout(0.3) | 56x15x32 | | | |
| Conv2D-4 (3x3), ReLU | 56x15x64 | 18,496 | 15,536,640 | 53,760 |
| GlobalMaxPool2D[1] | 1x64 | | | |
| Dense-1 | 1x64 | 4,160 | 4,160 | 64 |
| Dense-2 | 1x64 | 4,160 | 4,160 | 64 |
| Dense-3 | 1x1 | 65 | 65 | 1 |
| Total | | 45,697 | 188,153,025 | 2,315,649 |

[1] We used global average pooling in the binarized networks instead.

the weights, using PyTorch standard quantization mechanisms, with the BAM model trained using QAT. This combination resulted in task performance of $0.81 \pm 0.01$ test PCC, on par with the baseline. At the same time, the binary mixed-precision model reduces the required memory per input sample during inference approximately 25-fold, from 9.66 MB to 0.39 MB, assuming a theoretical 1-bit representation of the BAMs and an 8-bit representation of the input spectrogram. In conclusion, among the combinations we tested, a mixed-precision approach with BAMs and quantized weights seems most promising.

We wondered whether the above changes would already result in measurable computational gains, despite the fact that current deep learning libraries do not support mixed-precision quantization out of the box. To test this idea, we used off-the-shelf 8-bit PTQ in PyTorch to quantize the above model (Methods Sec. 2.2). Hence, in this setup, the 1-bit activations were wastefully represented with 8-bit tensors. We measured the inference wall-clock time of the 8-bit model running on a CPU, as PyTorch does not support running quantized models on GPUs. We found that the 8-bit quantized model showed reduced inference time by 56.3 % at comparable performance of $0.83 \pm 0.03$ test PCC to the baseline model. Thus, quantization has a measurable effect on inference time. However, this reduction by approximately $2\times$ is only useful as a proxy, as the underlying implementation does not reap the performance gain that could be achieved by exploiting the int8 $\times$ int1 mixed-precision quantization. Taking full advantage of this reduction will require special software and hardware support.

## 5. Discussion

Here, we investigated how to make CNN-based SQP models more efficient through quantization. To that end, we modified the DNSMOS architecture with BAMs and 8-bit PTQ of weights. We found that our mixed-precision quantization scheme did not lead to significant decreases in task performance, while offering substantial memory and computational gains. While we could not test how this extreme form of quantization translated into performance gains due to lack of suitable software support, we did measure a $2\times$ speed-up of inference time on CPU when we used off-the-shelf 8-bit PTQ of the whole model. However, we expect that larger performance gains are possible on dedicated accelerators and software frameworks with full BAM support.

While these results are promising, the present study has several limitations. First, since we derived our training data from a DNS dataset, it mostly comprises low-SNR samples, why our PESQ label distribution is considerably skewed towards low values. Thus, for our system to be applicable in real-world scenarios, a training set with a more realistic distribution of ground-truth labels might prove necessary. Second, our study is limited to a convolutional architecture and does not consider more sophisticated, and therefore more computationally demanding models [8, 12]. While such models may also benefit from BAMs, we deemed them less well suited for continuous resource-efficient SQP. Third, our weight quantization is limited to an 8-bit representation and only applied after training. In future work, we intend to both investigate sub-8-bit quantization and systematically survey QAT for different weight-quantization levels. Finally, our efficiency evaluation was limited by the lack of readily available hardware accelerators supporting BAMs. An in-depth evaluation on mixed-precision hardware is left for future work.

In summary, our results indicate considerable efficiency gains in SQP by combining BAMs and non-binary weights on accelerators supporting mixed-precision arithmetic.

## 6. Acknowledgements

This work was supported by the Swiss National Science Foundation (Grant No. PCEFP3_202981), EU's Horizon Europe Research and Innovation Programme (Grant Agreement No. 101070374, CONVOLVE) funded through SERI (Ref. 1131-52302), and the Novartis Research Foundation.

## 7. References

[1] E. Tzinis, Z. Wang, and P. Smaragdis, "Sudo rm -rf: Efficient Networks for Universal Audio Source Separation," in *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2020, pp. 1–6.

[2] D. Bralios, E. Tzinis, G. Wichern, P. Smaragdis, and J. L. Roux, "Latent Iterative Refinement for Modular Source Separation," in *ICASSP 2023 - 2023 IEEE International Conference on*

*Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–5.

[3] I. Fedorov, M. Stamenovic, C. Jensen, L.-C. Yang, A. Mandell, Y. Gan, M. Mattina, and P. N. Whatmough, "TinyLSTMs: Efficient Neural Speech Enhancement for Hearing Aids," in *Interspeech 2020*, Oct. 2020, pp. 4054–4058.

[4] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 2, May 2001, pp. 749–752 vol.2.

[5] 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, Sep. 2011.

[6] J. L. 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)*, May 2019, pp. 626–630.

[7] C. K. Reddy, E. Beyrami, J. Pool, R. Cutler, S. Srinivasan, and J. Gehrke, "A scalable noisy speech dataset and online subjective test framework," *Proc. Interspeech 2019*, pp. 1816–1820, 2019.

[8] S.-w. Fu, Y. Tsao, H.-T. Hwang, and H.-M. Wang, "Quality-Net: An End-to-End Non-intrusive Speech Quality Assessment Model Based on BLSTM," in *Interspeech 2018*. ISCA, Sep. 2018, pp. 1873–1877.

[9] M. K. Jayesh, M. Sharma, P. Vonteddu, M. A. B. Shaik, and S. Ganapathy, "Transformer Networks for Non-Intrusive Speech Quality Prediction," in *Interspeech 2022*. ISCA, Sep. 2022, pp. 4078–4082.

[10] M. Yu, C. Zhang, Y. Xu, S.-X. Zhang, and D. Yu, "MetricNet: Towards Improved Modeling For Non-Intrusive Speech Quality Assessment," in *Proc. Interspeech 2021*, 2021, pp. 2142–2146.

[11] R. E. Zezario, S.-W. Fu, C.-S. Fuh, Y. Tsao, and H.-M. Wang, "STOI-Net: A Deep Learning based Non-Intrusive Speech Intelligibility Assessment Model," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Dec. 2020, pp. 482–486.

[12] A. Kumar, K. Tan, Z. Ni, P. Manocha, X. Zhang, E. Henderson, and B. Xu, "Torchaudio-Squim: Reference-Less Speech Quality and Intelligibility Measures in Torchaudio," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–5.

[13] A. A. Catellier and S. D. Voran, "Wideband Audio Waveform Evaluation Networks: Efficient, Accurate Estimation of Speech Qualities," *IEEE Access*, vol. 11, pp. 125 576–125 592, 2023.

[14] R. E. Zezario, S.-W. Fu, F. Chen, C.-S. Fuh, H.-M. Wang, and Y. Tsao, "Deep Learning-Based Non-Intrusive Multi-Objective Speech Assessment Model With Cross-Domain Features," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 54–70, 2023.

[15] C. K. A. Reddy, V. Gopal, and R. Cutler, "Dnsmos: A Non-Intrusive Perceptual Objective Speech Quality Metric to Evaluate Noise Suppressors," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 6493–6497.

[16] G. Mittag, B. Naderi, A. Chehadi, and S. Möller, "NISQA: A Deep CNN-Self-Attention Model for Multidimensional Speech Quality Prediction with Crowdsourced Datasets," in *Interspeech 2021*, Aug. 2021, pp. 2127–2131.

[17] P. Manocha, B. Xu, and A. Kumar, "NORESQA: A Framework for Speech Quality Assessment using Non-Matching References," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 22 363–22 378.

[18] M. Elminshawi, S. R. Chetupalli, and E. A. P. Habets, "Slim-Tasnet: A Slimmable Neural Network for Speech Separation," in *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. New Paltz, NY, USA: IEEE, Oct. 2023, pp. 1–5.

[19] R. Miccini, A. Zniber, C. Laroche, T. Piechowiak, M. Schoeberl, L. Pezzarossa, O. Karrakchou, J. Sparsø, and M. Ghogho, "Dynamic nsNET2: Efficient Deep Noise Suppression with Early Exiting," in *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2023, pp. 1–6.

[20] S. Chen, Y. Wu, Z. Chen, T. Yoshioka, S. Liu, J. Li, and X. Yu, "Don't Shoot Butterfly with Rifles: Multi-Channel Continuous Speech Separation with Early Exit Transformer," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 6139–6143.

[21] A. Li, C. Zheng, L. Zhang, and X. Li, "Learning to Inference with Early Exit in the Progressive Speech Enhancement," in *2021 29th European Signal Processing Conference (EUSIPCO)*, Aug. 2021, pp. 466–470.

[22] L. Sun, J. Du, X. Zhang, T. Gao, X. Fang, and C.-H. Lee, "Progressive Multi-Target Network Based Speech Enhancement with Snr-Preselection for Robust Speaker Diarization," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 7099–7103.

[23] Y. Bengio, N. Léonard, and A. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," Aug. 2013, arXiv:1308.3432 [cs]. [Online]. Available: http://arxiv.org/abs/1308.3432

[24] K.-H. Ho, E.-L. Yu, J.-W. Hung, and B. Chen, "NAaLOSS: Rethinking the Objective of Speech Enhancement," in *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*. Rome, Italy: IEEE, Sep. 2023, pp. 1–6.

[25] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1," Mar. 2016, arXiv:1602.02830 [cs]. [Online]. Available: http://arxiv.org/abs/1602.02830

[26] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, Nov. 2019.

[27] 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 (CVPR)*, 2018, pp. 2704–2713.

[28] F. Zenke and S. Ganguli, "SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks," *Neural Computation*, vol. 30, no. 6, pp. 1514–1541, Jun. 2018.

[29] H. Dubey, A. Aazami, V. Gopal, B. Naderi, S. Braun, R. Cutler, A. Ju, M. Zohourian, M. Tang, M. Golestaneh, and R. Aichner, "ICASSP 2023 Deep Noise Suppression Challenge," *IEEE Open Journal of Signal Processing*, pp. 1–13, 2024.

[30] J. Timcheck, S. B. Shrestha, D. B. D. Rubin, A. Kupryjanow, G. Orchard, L. Pindor, T. Shea, and M. Davies, "The Intel neuromorphic DNS challenge," *Neuromorphic Computing and Engineering*, vol. 3, no. 3, p. 034005, Aug. 2023.

[31] T. Yep, "torchinfo," Mar. 2020. [Online]. Available: https://github.com/TylerYep/torchinfo

[32] C. K. Reddy, V. Gopal, R. Cutler, E. Beyrami, R. Cheng, H. Dubey, S. Matusevych, R. Aichner, A. Aazami, S. Braun, P. Rana, S. Srinivasan, and J. Gehrke, "The INTERSPEECH 2020 Deep Noise Suppression Challenge: Datasets, Subjective Testing Framework, and Challenge Results," in *Interspeech 2020*. ISCA, Oct. 2020, pp. 2492–2496.