Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

Efficient Streaming Speech Quality Prediction with Spiking Neural Networks

Mattias Nilsson[1], *Riccardo Miccini*[2,3], *Julian Rossbroich*[1,4], *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

mattias.nilsson@fmi.ch, friedemann.zenke@fmi.ch

Abstract

As speech processing systems become more ubiquitous, the need for real-time, efficient speech quality prediction (SQP) is growing. Conventional artificial neural networks (ANNs) offer strong prediction performance but can be computationally demanding, which limits their deployment on mobile and edge devices. Spiking neural networks (SNNs) present a promising alternative for ultra-low-power, streaming inference due to their sparse activity and event-driven processing. However, their potential for SQP remains largely unexplored. This article introduces deep convolutional SNNs for SQP and evaluates their performance against state-of-the-art ANN models. Our results show that SNNs achieve comparable accuracy while significantly reducing computational cost. These findings highlight the potential of SNNs to enable real-time, energy-efficient SQP in resource-constrained settings.

Index Terms: speech quality prediction, spiking neural networks, neuromorphic computing, edge computing

1. Introduction

Real-time speech quality prediction (SQP) is essential for applications ranging from telecommunications monitoring to adaptive speech enhancement. Traditional artificial neural networks (ANNs) have demonstrated strong performance in SQP but come with high computational costs, making them impractical for mobile and edge deployment. Recent advances in dynamic neural network architectures [1] have introduced methods such as early exiting [2–4], dynamic channel pruning [5], or mixture-of-experts [6] to optimize computational efficiency or adapt to different acoustic scenarios. This highlights the potential for using real-time SQP to dynamically adjust model complexity. However, these dynamic network approaches remain constrained by the computational overhead of existing SQP methods, which can offset their efficiency gains.

Previous work on models for efficient SQP have relied on either 1D temporal or 2D time–frequency convolutional architectures [7–9]. While effectively using methods like depthwise separable convolutions or binary activation maps to create more lightweight and efficient models, these models process entire temporal windows for each prediction update. This sliding window pattern gives rise to inherent computational redundancy in streaming scenarios, which limits the efficiency for real-time applications. Spiking neural networks (SNNs) [10] offer a fundamentally different approach, leveraging event-driven processing and sparse activations to achieve energy-efficient streaming inference. While SNNs have shown promise for low-power audio processing [11–14], their potential for SQP remains largely unexplored.

Here, we present a deep convolutional spiking neural network (CSNN) architecture for efficient, streaming real-time SQP and compare its performance against conventional ANN-based models. By training our model with different levels of activation sparsity, we show robust performance at up to 99 % sparsity. Finally, we present estimates of how our CSNN reduces computational operations compared to conventional ANNs on suitable hardware that supports sparsity. Our findings demonstrate that CSNNs can achieve comparable predictive accuracy while significantly reducing computational cost, paving the way for high-performance, power-efficient SQP systems.

2. Methods

2.1. Model

2.1.1. Preprocessing and Spike Encoding

The model (Fig. 1) takes one log-mel power spectrogram frame per time-step as input. We computed the spectrograms with 120 mel filters using a short-time Fourier transform (STFT) window size of 40 ms with 20 ms hop length. This corresponds to 249 frames and corresponding SNN time-steps per 5-s sample. The frequency bands are fanned out into 16 copies along the convolutional channel dimension and fed one-to-one as continuous-valued currents to adaptive leaky integrate-and-fire (adLIF) neurons [15], which are included in training and thereby learn multiple spike encodings for each band. The adLIF neuron model has intrinsic stabilizing properties by virtue of spike-frequency adaptation and has been demonstrated to reduce the need for explicit data normalization techniques [16]. We used no data normalization in this work.

2.1.2. Convolutional SNN Architecture

The encoder layer is followed by a number of 1D convolutional layers that consist of standard leaky integrate-and-fire (LIF) neurons. Whereas a conventional 2D CNN would use convolutions over time–frequency space, our CSNN uses 1D convolutions over the frequency dimension, while temporal information is integrated by neuron and synapse dynamics. Thereby, the model is intrinsically suited for streaming processing. We use a kernel size of five with a stride of two for downsampling, instead of max-pooling, to reduce computational cost. In the case of recurrent convolutions, on the other hand, a stride of one is used to preserve dimensions. Finally, the convolutional layers are followed by a non-spiking leaky integrator (LI) neuron that provides a continuous readout signal from the network. All adLIF, LIF, and LI units in the network have dynamic, exponentially decaying neurons and synapses. Their time-constants are unit-specific and learnable [17, 18], and so is the adLIF adaptation parameter.

10.21437/Interspeech.2025-2269

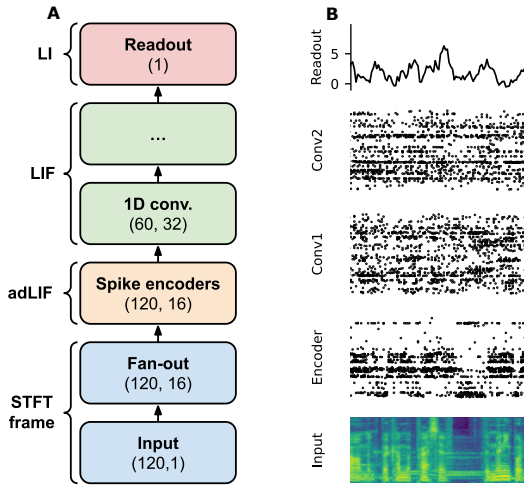

Figure 1: ***(A) Convolutional SNN architecture for speech quality prediction.*** *The dimensions correspond to the processing of one frame in one SNN time-step. The numbers in parentheses indicate the spectral and channel output dimensions, respectively. The number of LIF neurons in each convolutional layer corresponds to its number of output activations.* ***(B) Example network activity during 2 s.***

2.2. Dataset

As our work targets speech enhancement settings, we used data from the DNS Challenge 2020 [19], generated using the provided synthesization scripts with configuration parameters matching those used for TorchAudio-Squim [20]—i.e., a signal-to-noise ratio (SNR) range of -15 to 25 dB and sample length of 5 s. However, to constrain training time, we limited the amount of data to 50 h. Furthermore, as in [20], we preprocessed the data with various levels of noise suppression. For this, we augmented each sample by either leaving it unprocessed or processing it with one of four configurations of an early-exiting speech enhancement model [4].

In the absence of human mean opinion scores (MOS), we labeled the data with the perceptual evaluation of speech quality (PESQ) metric [21], for which both the clean and degraded samples are used to compute an estimate of perceived speech quality. We created a validation set by holding out the data of 5 % of randomly selected speakers from the training set. For testing, we used the synthetic reverberation-free test data of the DNS Challenge 2020 dataset. Since this test set is relatively small, consisting of 150 samples here truncated to 5 s each, we increased its effective size by including all four levels of noise suppression for each sample to create a more comprehensive test set. The PESQ distributions of our training and test sets are presented in Fig. 2.

2.3. Training

The CSNNs were trained to minimize the mean squared error (MSE) between the predicted and true PESQ values using surrogate gradient descent [22]. As there is only a single PESQ score per sample, we defined the mean-over-time of the continuous network readout as its prediction. We used a learning rate of 3×10^{-3}, a batch size of 128, the SuperSpike surrogate gradient [23] with a steepness of $\beta = 20$, and a custom modification of the SMORMS3 optimizer [24, 25]. The models were initialized

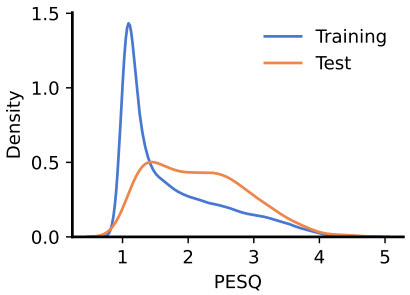

Figure 2: ***PESQ label distributions of the training and test sets.***

in the fluctuation-driven regime [25]. To avoid overfitting, we used early stopping based on the Pearson correlation coefficient (PCC) on validation data to end training after 25 epochs without improvement. All training runs finished within 300 epochs. We also used dropout values in the convolutional layers of 0.4 for the recurrent model and 0.15 for the feedforward model, determined through parameter sweeps. All SNN simulations were done with a 20-ms time-step, matching the hop length of the spectrograms. We ran all SNN simulations with Python 3.10.12 using custom code based on the Stork SNN simulator [25] and PyTorch [26]. The code will be made available upon publication at `https://github.com/fmi-basel/spiking-sqp`.

2.3.1. Activity Regularization

To encourage sparse neuronal activity, we employed an L2 upper-bound regularization term on population-level average spike-counts [27]. Similarly, to avoid silent features during training, we used an L2 lower-bound term to ensure that every feature map spikes at least once per sample.

2.4. Evaluation

We evaluated our models in terms of the PCC and MSE against two existing lightweight ANNs for SQP: DNSMOS [28], a CNN, and Quality-Net [29], an LSTM-based architecture. While MOSLight [7] has shown promising results for lightweight SQP using depthwise separable temporal convolutions, we focus our comparison on Quality-Net implemented with unidirectional LSTM as the most relevant baseline for streaming data scenarios. We also included DNSMOS, a lightweight model with non-causal processing widely used as a benchmark in SQP and speech enhancement, as a proxy for an upper performance bound. The baseline ANNs were trained using the Adam optimizer with standard parameters, a learning rate of 10^{-3}, and a scheduler decreasing the learning rate by a factor of 0.9 after every five epochs without improvements on the validation loss. Beyond that, we used the same training setup as described for the CSNNs.

2.5. Efficiency Metrics

We estimated the computational cost of our CSNNs and the baseline models in terms of effective floating-point operations (EFLOPs) [30] that do not count multiplications or additions with zero, which can be skipped on suitable sparsity-supporting hardware. We categorize the total operations of a model into *static* and *dynamic EFLOPs*, which are, respectively, independent and dependent on activation sparsity. For SNNs on digital hardware, the static EFLOPs consist of the operations required

for neuron and synapse state updates at each time-step, while the dynamic EFLOPs consist of the sparsity-dependent operations of accumulating weighted input spikes in each neuron. The number of static EFLOPs per neuron and time-step are five for the LIF and LI neurons and 11 for the adLIF neurons. These numbers neglect the spike-triggered reset, as the activation sparsity of all our CSNNs is above 90 %. We measured dynamic EFLOPs in SNNs as described above using the NeuroBench evaluation framework for neuromorphic computing [31].

Similarly, we used NeuroBench to measure effective MAC operations in DNSMOS, which has some degree of activation sparsity due to zero-activations in ReLU units. For Quality-Net, the LSTM layer uses tanh and sigmoid activations, resulting in no activation sparsity. Since the LSTM layer accounts for virtually all the computational load in Quality-Net, and since NeuroBench does not presently support LSTM layers, we used torchinfo [32] to calculate the number of FLOPs in Quality-Net without considering activation sparsity.

3. Results

To evaluate SNNs for SQP, we developed two CSNN architectures. First, we implemented a feedforward version (ff-CSNN) inspired by DNSMOS [28], having four convolutional layers with the same spectral and channel dimensions (32-32-32-64), but without subsequent dense layers as in DNSMOS, since they did not improve CSNN performance. Second, we investigated "r-CSNN" with added recurrent connections in the convolutional layers of our model. This variation had only two convolutional layers (32-64), since we found no benefit in additional depth.

3.1. Training and Performance

We trained both architectures to predict PESQ on the DNS Challenge 2020 dataset. Both models showed similar convergence behavior (Fig. 3), with the r-CSNN doing marginally better on validation data, and the ff-CSNN exhibiting higher fluctuations in the validation loss. The validation PCC of both architectures plateaued after around 150 epochs, upon which we used early stopping. The r-CSNN performed better than ff-CSNN, almost on par with Quality-Net (Table 1). These results demonstrate that SNNs can achieve SQP performance competitive to a model based on a unidirectional LSTM, while both are outperformed by the non-causal processing of DNSMOS.

3.2. Activation Sparsity

To further improve computational efficiency on neuromorphic hardware supporting sparse processing, we sought to find an optimal trade-off between activation sparsity and performance. To that end, we trained the CSNN models with population spike-count regularization (Sec. 2.3.1) at a number of different upper-bound levels (Fig. 4). We found a default activation sparsity of about 94 % for both models when sparsity was not enforced. The activation sparsity of both models could be increased to about 99 % before their performance started to deteriorate sharply. This difference corresponds to a decrease in average neuronal firing rates by about one order of magnitude, from ≈ 3 to 0.3 Hz, before the sharp performance drop. Finally, the r-CSNN performed slightly better than the ff-CSNN overall and was more robust at high levels of sparsity.

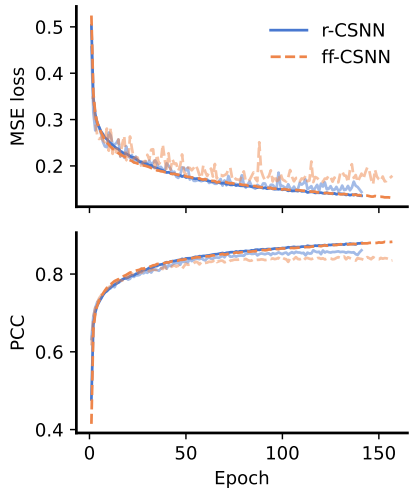

Figure 3: ***Training performance of CSNN models.*** *Dark and light lines indicate training and validation data, respectively.*

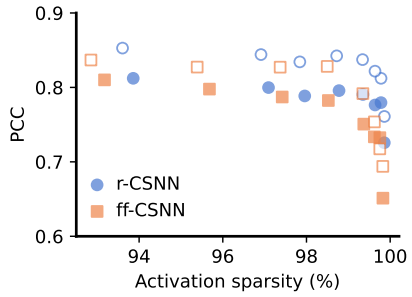

Figure 4: ***Performance of CSNN models at different levels of activation sparsity.*** *Filled and unfilled markers indicate test and validation data, respectively.*

3.3. Computational Efficiency

We estimated the computational cost of the models in terms of EFLOPs as described in Sec. 2.5. Importantly, both CSNN variants achieve significant efficiency gains over the baseline models (Table 1). The ff-CSNN and r-CSNN had similar computational costs, using roughly half the number of EFLOPs per sample compared to Quality-Net, which in turn used roughly half compared to DNSMOS. Importantly, this difference to DNSMOS does not yet include the computational overhead due to redundancy of its sliding window in the streaming data setting, which would be added on top. Overall, the computational load of both CSNNs was split roughly evenly between the static and dynamic components. Using the operating point at 99.3 % activation sparsity of the r-CSNN brought down its dynamic EFLOPs by about $5\times$ at the price of a moderate performance degradation, which may be acceptable for resource-constrained applications (Table 1). In the sparse operating regime, the static EFLOPs were the dominant component, determining the order of magnitude for the total EFLOPs.

Table 1: ***Task performance and computational cost on test set.*** *Performance metrics are reported as mean ± standard deviation across five runs. The numbers of EFLOPs are per 5-s input sample, with dynamic EFLOPs averaged over the test set. Bold font indicates best results.*

	Model	Streaming	Spiking	PCC ↑	MSE ↓	EFLOPs			Params.
						Static	Dynamic	Total	
Baselines	DNSMOS (CNN)	✗	✗	$\mathbf{0.865 \pm 0.008}$	0.272 ± 0.024	–	93.2 M	93.2 M	46 k
	Quality-Net[1] (LSTM)	✓	✗	0.830 ± 0.014	0.203 ± 0.025	44.2 M	–	44.2 M	96 k
Our models	r-CSNN	✓	✓	0.810 ± 0.002	$\mathbf{0.179 \pm 0.005}$	10.0 M	11.3 M	21.4 M	56 k
	r-CSNN (99.3 % sparsity)	✓	✓	0.782 ± 0.009	0.206 ± 0.006	10.0 M	2.25 M	**12.3 M**	56 k
	ff-CSNN	✓	✓	0.798 ± 0.010	0.206 ± 0.013	10.1 M	8.57 M	18.7 M	**39 k**

[1] Causal reimplementation based on unidirectional LSTM.

4. Discussion

Here, we developed a deep CSNN architecture for efficient streaming SQP. Specifically, we investigated two versions with recurrent and feedforward convolutional layers, respectively. The r-CSNN performs comparatively to Quality-Net with unidirectional LSTM, while reducing the amount of estimated computational operations on neuromorphic hardware by half or more, depending on the degree of activation sparsity. We characterized the trade-off between activation sparsity and performance in the CSNNs, and found that it can be increased from the default of 94 % with a modest impact on performance to about 99 %, after which performance drops substantially. These findings demonstrate that SNNs can achieve competitive SQP performance while offering significant efficiency advantages through activation sparsity. Furthermore, our architectures avoid the use of non-conventional neuron models and spike delays, which could complicate training and hardware deployment.

While our CSNNs natively support streaming inference, they were trained in a conventional SQP setup. This means that labels are provided at the level of samples, i.e., with a coarse temporal resolution, as SQP is typically based on human MOS labeling and objective metrics like PESQ. Here, we simply let the mean-over-time per sample of the continuous network readout represent its PESQ estimate. Future work targeting streaming SQP could explore inclusion of time-varying objective metrics, such as frame-wise signal-to-distortion ratio (SDR), to evaluate and improve the precision of predictions on finer timescales.

Our efficiency analysis revealed that neuronal state updates dominate the computational cost in sparse CSNNs. This suggests that future work should explore reducing the density of neuronal state computations. This could for instance be achieved by using approximation schemes for neuronal state updates [33, 34]. However, this only applies to digital hardware implementations of SNNs, as on mixed-signal neuromorphic hardware, analog electronics are used to emulate neuronal dynamics at low cost [35, 36]. Such hardware implementations of our models could be substantially more efficient than what the digital-hardware estimates we have presented imply.

In summary, our findings suggest that SNNs are promising architectures for SQP in resource-constrained settings, such as edge devices and real-time applications.

5. Acknowledgements

We thank Julia Gygax for support with SNN simulations and training, and all members of the Zenke Lab for constructive comments and discussions. 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.

6. References

[1] Y. Han *et al.*, "Dynamic Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7436–7456, Nov. 2022.

[2] A. Li *et al.*, "Learning to Inference with Early Exit in the Progressive Speech Enhancement," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, Aug. 2021, pp. 466–470.

[3] S. Kim and M. Kim, "Bloom-Net: Blockwise Optimization for Masking Networks Toward Scalable and Efficient Speech Enhancement," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2022, pp. 366–370.

[4] R. Miccini *et al.*, "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.

[5] ——, "Scalable Speech Enhancement With Dynamic Channel Pruning," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2025, pp. 1–5.

[6] R. E. Zezario *et al.*, "Speech Enhancement with Zero-Shot Model Selection," in *2021 29th European Signal Processing Conference (EUSIPCO)*, Aug. 2021, pp. 491–495.

[7] Z. Li and W. Li, "MOSLight: A Lightweight Data-Efficient System for Non-Intrusive Speech Quality Assessment," in *INTERSPEECH 2023*. ISCA, Aug. 2023, pp. 5386–5390.

[8] M. Nilsson *et al.*, "Resource-Efficient Speech Quality Prediction through Quantization Aware Training and Binary Activation Maps," in *Interspeech 2024*. ISCA, Sep. 2024, pp. 2975–2979.

[9] F. Cumlin *et al.*, "DNSMOS Pro: A Reduced-Size DNN for Probabilistic MOS of Speech," in *Interspeech 2024*. ISCA, Sep. 2024, pp. 4818–4822.

[10] J. K. Eshraghian *et al.*, "Training Spiking Neural Networks Using Lessons From Deep Learning," *Proceedings of the IEEE*, vol. 111, no. 9, pp. 1016–1054, Sep. 2023.

[11] H. Bos and D. Muir, "Sub-mW Neuromorphic SNN Audio Processing Applications with Rockpool and Xylo," in *Embedded Artificial Intelligence*. River Publishers, 2023.

[12] S. B. Shrestha *et al.*, "Efficient Video and Audio Processing with Loihi 2," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2024, pp. 13 481–13 485, iSSN: 2379-190X.

[13] T. Sun and S. Bohté, "DPSNN: spiking neural network for low-latency streaming speech enhancement," *Neuromorphic Computing and Engineering*, vol. 4, no. 4, p. 044008, Dec. 2024.

[14] X. Hao *et al.*, "Toward Ultralow-Power Neuromorphic Speech Enhancement With Spiking-FullSubNet," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2025.

[15] B. Cramer *et al.*, "Surrogate gradients for analog neuromorphic computing," *Proceedings of the National Academy of Sciences*, vol. 119, no. 4, Jan. 2022.

[16] M. Baronig *et al.*, "Advancing Spatio-Temporal Processing in Spiking Neural Networks through Adaptation," Aug. 2024, arXiv:2408.07517 [cs].

[17] W. Fang *et al.*, "Incorporating Learnable Membrane Time Constant to Enhance Learning of Spiking Neural Networks," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 2641–2651.

[18] N. Perez-Nieves *et al.*, "Neural heterogeneity promotes robust learning," *Nature Communications*, vol. 12, no. 1, p. 5791, Oct. 2021.

[19] C. K. Reddy *et al.*, "The INTERSPEECH 2020 Deep Noise Suppression Challenge: Datasets, Subjective Testing Framework, and Challenge Results," in *Interspeech 2020*. ISCA, Oct. 2020, pp. 2492–2496.

[20] A. Kumar *et al.*, "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.

[21] A. Rix *et al.*, "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.

[22] 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.

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

[24] S. Funk, "RMSprop loses to SMORMS3 - Beware the Epsilon!" [Online]. Available: https://sifter.org/~simon/journal/20150420.html

[25] J. Rossbroich, J. Gygax, and F. Zenke, "Fluctuation-driven initialization for spiking neural network training," *Neuromorphic Computing and Engineering*, vol. 2, no. 4, p. 044016, Dec. 2022.

[26] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[27] F. Zenke and T. P. Vogels, "The Remarkable Robustness of Surrogate Gradient Learning for Instilling Complex Function in Spiking Neural Networks," *Neural Computation*, vol. 33, no. 4, pp. 899–925, Mar. 2021.

[28] 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.

[29] S.-w. Fu *et al.*, "Quality-Net: An End-to-End Non-intrusive Speech Quality Assessment Model Based on BLSTM," in *Interspeech 2018*. ISCA, Sep. 2018, pp. 1873–1877.

[30] S. Narduzzi *et al.*, "Optimizing The Consumption Of Spiking Neural Networks With Activity Regularization," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 61–65.

[31] J. Yik *et al.*, "The neurobench framework for benchmarking neuromorphic computing algorithms and systems," *Nature Communications*, vol. 16, no. 1, p. 1545, Feb. 2025.

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

[33] R. Wang *et al.*, "A stochastic approach to STDP," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2016, pp. 2082–2085.

[34] S. Eissa, S. Stuijk, and H. Corporaal, "Hardware Approximation of Exponential Decay for Spiking Neural Networks," in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, Jun. 2021, pp. 1–4.

[35] C. Pehle *et al.*, "The BrainScaleS-2 Accelerated Neuromorphic System With Hybrid Plasticity," *Frontiers in Neuroscience*, vol. 16, Feb. 2022.

[36] O. Richter *et al.*, "DYNAP-SE2: a scalable multi-core dynamic neuromorphic asynchronous spiking neural network processor," *Neuromorphic Computing and Engineering*, vol. 4, no. 1, p. 014003, Jan. 2024.