DPSNN: spiking neural network for low-latency streaming speech enhancement

To cite this article: Tao Sun and Sander Bohté 2024 *Neuromorph. Comput. Eng.* **4** 044008

NEUROMORPHIC
Computing and Engineering

OPEN ACCESS

RECEIVED
14 August 2024

REVISED
25 October 2024

ACCEPTED FOR PUBLICATION
18 November 2024

PUBLISHED
20 December 2024

PAPER

DPSNN: spiking neural network for low-latency streaming speech enhancement

Tao Sun[1,*] **and Sander Bohté**[1,2]

[1] Machine Learning Group, CWI, Amsterdam, The Netherlands
[2] Cognitive and Systems Neuroscience Group, University of Amsterdam, Amsterdam, The Netherlands
[*] Author to whom any correspondence should be addressed.

E-mail: tao.Sun@cwi.nl and sbohte@cwi.nl

Keywords: spiking neural networks, speech enhancement, noise suppression, low latency, streaming

Abstract

Speech enhancement improves communication in noisy environments, affecting areas such as automatic speech recognition (ASR), hearing aids, and telecommunications. With these domains typically being power-constrained and event-based, and often requiring low latency, neuromorphic algorithms–particularly spiking neural networks (SNNs)–hold significant potential. However, current effective SNN solutions require a long temporal window to calculate Short Time Fourier Transforms (STFTs) and thus impose substantial latency, typically around 32 ms, which is too long for applications such as hearing aids. Inspired by the Dual-Path Recurrent Neural Network (DPRNN) in deep neural networks (DNNs), we develop a two-phase time-domain streaming SNN fframework for speech enhancement, named *Dual-Path Spiking Neural Network (DPSNN)*. DPSNNs achieve low latency by replacing the STFT and inverse STFT (iSTFT) in traditional frequency-domain models with a learned convolutional encoder and decoder. In the DPSNN, the first phase uses Spiking Convolutional Neural Networks (SCNNs) to capture temporal contextual information, while the second phase uses Spiking Recurrent Neural Networks (SRNNs) to focus on frequency-related features. In addition, threshold-based activation suppression, along with L_1 regularization loss, is applied to specific non-spiking layers in DPSNNs to further improve their energy efficiency. Evaluating on the Voice Cloning Toolkit (VCTK) Corpus and Intel N-DNS Challenge dataset, our approach demonstrates excellent performance in speech objective metrics, along with the very low latency (approximately 5 ms) required for applications like hearing aids.

1. Introduction

Speech enhancement refines and clarifies spoken communication in the presence of undesirable noisy conditions [1]. Beyond automatic speech recognition (ASR) and speaker recognition, effective speech enhancement is also vital in domains such as hearing aids and mobile telecommunications. Machine learning methods, particularly deep neural networks (DNNs), have emerged as the main approach for speech enhancement in the last decade [2–9].

For many speech enhancement applications, low-latency processing is critical to ensure effective speech communication, as in the case of hearing aids [10]. Latency, also called processing latency, is defined as the delay between the input of an audio signal and the corresponding output of the processed signal (figure 1(a)). Latency is decomposed into two components, algorithmic latency and hardware latency, where algorithmic latency refers to the latency caused by algorithmic constraints, while hardware latency denotes the time needed by the hardware to process an input unit [11]. Current DNN-based speech enhancement solutions, particularly time-domain DNN models that directly handle and predict waveform signals, have achieved both high accuracy and low algorithmic latency simultaneously [10]. However, due to their large network sizes, these DNN solutions are usually energetically costly, limiting their applicability within the many power-constrained environments [12].

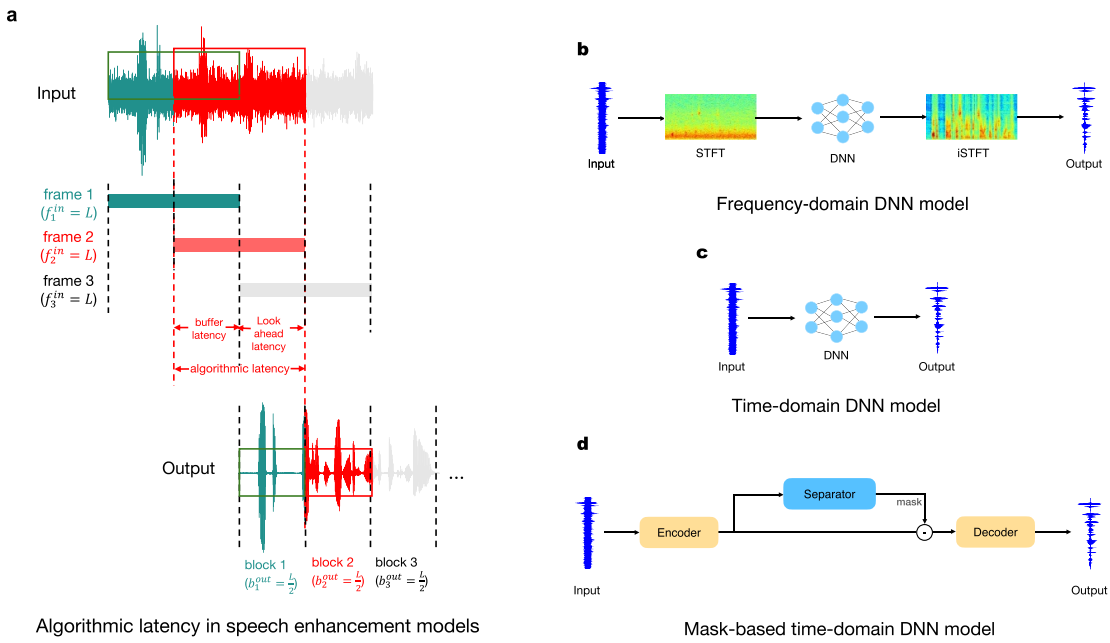

Figure 1. (a) The algorithmic latency in frame-based models consists of a buffering latency and a look-ahead latency. The buffering latency matches the frame-shift length (i.e., block size), whereas the look-ahead latency results from the extra look-ahead within a frame, typically used to provide additional processing context to improve performance. (b) A frequency-domain DNN model transforms a noisy audio signal to its T-F representation by the STFT and then fed it into a neural network. (c) Inputs and outputs to the time-domain DNN models are both time-domain signals. (d) Mask-based time-domain DNN models commonly adopt an encoder-separator-decoder architecture.

The development of spiking neural networks (SNNs) and corresponding neuromorphic hardware for speech enhancement is primarily driven by their potential for energy efficiency. The temporal characteristics of signal processing in speech enhancement align well with the capabilities of SNNs to manage dynamic, time-dependent tasks. Enhanced energy efficiency would extend battery life and enable smaller form factors for speech enhancement devices such as headsets, earbuds, hearing aids, and cochlear implants [12].

Early DNN-based speech enhancement solutions typically worked in the frequency-domain, where noisy audio signals are first transformed into Time-Frequency (T-F) representations through the Short Time Fourier Transform (STFT) and then fed into a neural network (figure 1(b)). During STFT, audio signals are divided into overlapping frames, each of which is then transformed into frequency forms. A frequency-based solution typically takes an individual frame as input and produces a block as output. As shown in the figure 1(a), the length of an input frame is the sum of the length of an output block and a future context. This setting implies that the production of a block is contingent upon the analysis of the entire frame. As a result, the algorithmic latency to output a block is equal to duration of a frame, comprising both the block duration (buffering latency) and the duration of the future context (look-ahead latency) [11]. Shorter frames reduce latency; however, frame length impacts both the time and frequency resolutions of a T-F representation: using short frames decreases frequency resolution, making it difficult to distinguish closely spaced frequency components, as needed for speech enhancement [13].

To balance the trade-off between the time and frequency resolutions and thus ensure enhancement performance, frequency-domain DNN solutions usually choose frame lengths as long as 32 ms [8, 11, 14]. This choice results in an algorithmic latency of 32 ms for those solutions. While such latency is acceptable for applications like audio communication [15], it is too high for scenarios like hearing aids. Due to the need for precise auditory-visual synchronization and to avoid mixing direct and processed sounds, hearing aid users have a very low tolerance for delay (typically only 20 to 30 ms) [10]. The Clarity speech enhancement challenge [16], targeting hearing aid, even has the latency requirement as low as 5 ms. However, simply reducing the frame length of existing frequency-domain speech enhancement models to meet such a low latency requirement will significantly impair their denoising performance [17], as demonstrated in one of our experiments and illustrated in figure 5. To achieve both low latency and satisfactory denoising results, substantial modifications to current frequency-domain enhancement models are needed [11]. In contrast, time-domain models (figure 1(c)), where both the input and output of the DNNs are time-domain waveforms, apply trained convolutional encoders and decoders in place of the STFT and inverse STFT (iSTFT) used in frequency-domain models, enabling them to process very short frames and achieve low-latency enhancement [7, 8]. Filters learned in the encoder of a time-domain model generally emphasize

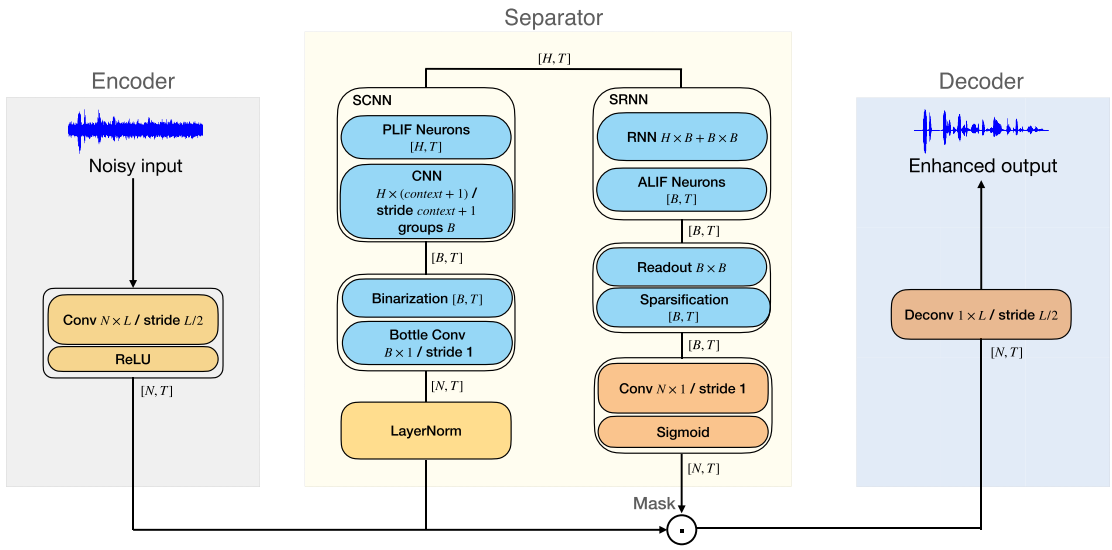

Figure 2. The proposed DPSNN adopts the encoder-separator-decoder architecture. The encoder uses convolutions to convert waveform signals into 2D feature maps, effectively replacing the function of STFT. In the separator, a 2D mask is calculated, primarily relying on the SCNN and SRNN modules that capture the temporal and frequency contextual information of the encoded feature maps, respectively. After applying the mask to the feature maps from the encoder, the decoder transforms the masked feature maps back to enhanced waveform signals.

low acoustic frequencies, which are crucial for speech intelligibility [8]. This ensures that the model's speech performance remains robust, even when using very short input frames [7, 8]. In the 2022 Clarity challenge, nearly all leading solutions were time-domain based DNN models [11].

Yet, to the best of our knowledge, current SNN-based speech enhancement solutions are in the frequency-domain, leading to long latencies (typically 32 ms) [18–20]. While these approaches have demonstrated satisfactory enhancement performance, their prolonged latencies make them impractical for applications requiring low latency. To tackle this issue, we here design a novel time-domain streaming SNN model for speech enhancement, achieving low latency through applying small frame sizes. Inspired from the success of Dual-Path Recurrent Neural Network (DPRNN) [9], our model applies a two-phase framework to capture rich contexts for sequence modeling. As illustrated in figure 2, the model, dubbed *Dual-Path Spiking Neural Network (DPSNN)*, first converts noisy signals into a two-dimensional (2D) feature map in the encoder using a convolution that functions similarly to the STFT, but with a significantly smaller kernel size, leading to much lower latency. A DPSNN then consists of two distinct modules, a Spiking Convolutional Neural Network (SCNN) module and a module based on Spiking Recurrent Neural Networks (SRNN) [21], to extract clean signals from noisy inputs. The initial phase of a DPSNN works with the SCNN module and captures contextual information along the temporal direction of the 2D feature map output by the encoder. The subsequent phase, via the SRNN module, then integrates context along the frequency direction of the 2D feature map. Additionally, our DPSNNs apply threshold-based and regularizer-based activation suppressions [22] to specific non-spiking layers, creating more sparse representations and thus enhancing energy-efficiency.

We conduct thorough evaluations of proposed DPSNNs using experiments on the Voice Cloning Toolkit (VCTK) corpus [23] and Intel N-DNS Challenge dataset [12], yielding excellent performance across latency, energy efficiency, and speech enhancement metrics. The source code for this work is available at https://github.com/tao-sun/dpsnn.

2. Related work

2.1. Time-domain solutions

Time-domain solutions can be divided into mapping-based and mask-based methods [24]. Mapping-based models typically apply an encoder-decoder architecture, directly mapping noisy inputs to denoised outputs [25–28]. Such models typically focus on offline applications, leveraging long contextual information (often multiple seconds in length), which results in significant latency [29]. Mask-based methods (figure 1(d)), and in particular the Time-Domain Audio Separation Network (TasNet) models [7–9], adopt an encoder-separator-decoder architecture and have demonstrated excellent performance in both separation and latency metrics [10, 11]. In the encoder of a mask-based method, a trained convolutional layer functions

similarly to an STFT to generate 2D spatiotemporal features, from which masks are computed in the separator. These masks are then multiplied with the spatiotemporal features to produce enhanced features. Lastly, the separated enhanced features are reverted to waveforms in the decoder that is comprised of a deconvolution layer. In the mask-based method, the kernel size of the convolution in the encoder is functionally equivalent to the frame size of the STFT. As such, the latency of a TasNet is determined by the kernel size used in its encoder. In the original TasNet framework [7], LSTM networks were employed within the separator to compute the mask. To alleviate the computational burden associated with such LSTM networks, Conv-TasNet [8] proposed the use of dilated convolutions in place of LSTMs within the separator, allowing for smaller kernel sizes and strides in convolutions within the encoder, enhancing its suitability for low-latency applications. To effectively capture long sequential inputs, DPRNNs were introduced [9], the separator of which consisted of two sequential RNNs processing shorter chunks of input signals: the first RNN operates within each chunk in parallel, integrating local frequency-related contexts, while the second RNN operates across chunks to capture long-term temporal information. The DPRNN serves as an inspiration for our DPSNN, where the SRNN is functionally similar to the first RNN of the DPRNN, while the SCNN is analogous to the second RNN.

2.2. Speech enhancement with SNNs

The Intel N-DNS challenge concentrates on speech enhancement tasks, recognizing them as a high-potential application domain for neuromorphic solutions. The Challenge winner, Spiking-FullSubNet [30], combines two frequency-domain approaches, using a full-band model and a sub-band model. The full-band model captures dependencies between frequency bands, while the sub-band model handles each band independently. Additionally, the Gated Spiking Neuron (GSN) is introduced in this model, where membrane potentials are calculated through time constants that could vary at each time step. Another competitive approach is the Spiking Structured State Space Model (SpikingS4) [19], which builds on the concept of structured state space modeling [31]. The Spiking-UNet [20], the only mapping-based method to date, combines the UNet [32] architecture with SNNs for single-channel noise reduction. Despite their strong performance in speech enhancement metrics, these SNN models, being frequency-domain methods, are unable to meet low-latency requirements in demanding application scenarios (e.g. hearing aids) without substantial modifications [11].

3. Methods

3.1. Problem setup

Speech enhancement improves the quality of speech signals by reducing or eliminating additive noise. The primary goal is to enhance the intelligibility and perceptual quality of speech in various real-world environments where noise interference is present [33].

One common approach to formally modeling speech enhancement involves the use of signal processing techniques to model the relationship between the observed noisy speech signal $y[n]$, the clean speech signal $s[n]$, and the additive noise signal $v[n]$. The noisy signal $y[n]$ can be expressed as the sum of the clean speech signal and the additive noise:

$$y[n] = s[n] + v[n]\,. \tag{1}$$

The objective of speech enhancement is then to estimate or reconstruct the clean speech signal $s[n]$ from the observed noisy signal $y[n]$. The enhanced speech signal $\tilde{s}[n]$ is written as:

$$\tilde{s}[n] = f(y[n])\,. \tag{2}$$

Speech enhancement is closely related to speech separation; the key difference being that speech enhancement aims to improve the quality of a noisy speech signal by removing or suppressing the unwanted noise, whereas speech separation focuses on separating individual speakers or sound sources from a mixture of multiple audio sources.

3.2. Spiking neural networks (SNNs)

SNNs typically use similar network topologies to artificial neural networks (ANNs), yet SNNs employ stateful, binary-valued spiking neurons as their computational units. Consequently, inference in SNNs unfolds iteratively across multiple time steps $t = 0, 1, \ldots, T$: at each time step t, the internal state of a neuron, represented by the membrane potential u_t, is influenced by incoming spikes from pre-synaptic neurons, if any, emitted at the previous time step $t-1$, along with the neuron's own membrane potential at the previous time step u_{t-1}. At time step t, a neuron emits a spike (indicated by an output of 1) when its membrane

potential reaches its threshold θ; otherwise, the neuron outputs 0. Notably, spiking generally occurs sparsely. This sparse, binary, and asynchronous communication among connected neurons allows SNNs to potentially achieve high energy efficiency.

Various spiking neuron models exist, ranging from the intricate and biologically detailed Hodgkin–Huxley model to the simplified leaky integrate-and-fire (LIF) neuron model [34]. For machine learning applications, SNNs mostly employ LIF spiking neurons, and variants thereof, due to their interpretability and computational efficiency. Resembling an RC circuit, the LIF neural model is mathematically represented as:

$$\tau_{\mathrm{m}} \frac{\mathrm{d}u}{\mathrm{d}t} = -(u - u_{\mathrm{rest}}) + RI, \tag{3}$$

where u_{rest} denotes the resting potential of the neuron, I expresses the input current, R is the membrane resistance, and τ_{m} represents the membrane time constant.

The discrete approximation of (3) can be expressed as

$$u_t = \left(1 - \frac{1}{\tau_{\mathrm{m}}}\right) u_{t-1} + \frac{1}{\tau_{\mathrm{m}}} (u_{\mathrm{rest}} + RI_t) \tag{4}$$

$$s_t = \Theta(u_t - \theta) \tag{5}$$

$$u_t = u_t (1 - s_t) + u_{\mathrm{rest}} s_t \tag{6}$$

where equation (4) describes subthreshold neural dynamics of a neuron; R is assumed to be 1 and $I_t = \Sigma_i w_i s^i_{t-1}$ is the input current from the pre-synaptic neurons, where w_i represents the weight connecting the neuron and its pre-synaptic neuron i and s^i_{t-1} indicates whether a pre-synaptic neuron i spikes in the last time step $t-1$; Θ in (5) is Heaviside step function deciding whether a neuron spikes; equation (6) calculates the final membrane potential of a neuron in a time step t.

For training SNNs, surrogate gradient methods [21, 35] enable straightforward supervised trainability. LIF neurons with learnable model parameters [21, 36] have demonstrated enhanced performance when used in SNNs. In [36], parametric LIF (PLIF) neurons are introduced where the time constant τ_{m} of a LIF neuron is learnable and shared by all neurons in one layer.

In [37], adaptive LIF (ALIF) neurons are proposed, where time constants τ_{m} is learnable for each individual LIF neuron. Additionally, for these ALIF neurons, the threshold of a neuron increases after spiking and then decays with a learnable time constant τ_{adp}. Dynamics in ALIF neurons for threshold θ and membrane potential u can be written as:

$$\theta = b_0 + \beta \eta_t \tag{7}$$

$$\eta_t = \rho \eta_{t-1} + (1 - \rho) s_{t-1} \tag{8}$$

$$u_t = \alpha u_{t-1} + (1 - \alpha) RI_t - s_{t-1}\theta, \tag{9}$$

where the dynamic threshold θ consists of a constant minimum b_0 and an adaptive component η_t, which evolves according to (8) and is scaled by the coefficient β; The variables $\alpha = \frac{1}{1+exp(-\tau_m)}$ and $\rho = \frac{1}{1+exp(-\tau_{adp})}$ express decay of the membrane potential u and threshold θ, respectively.

3.3. Architecture

As shown in figure 3(a), our speech enhancement model adopts the mask-based encoder-separator-decoder architecture. An encoder first takes overlapping waveform frames, each with a length of L, as inputs and converts these frames into features with N channels through a one-dimensional (1D) convolution and a ReLU activation function. Those features, aligned together, then form a 2D feature map. Note that the input frame duration determines the minimum frequency of input speech signals that the encoder can capture (e.g., 200 Hz for an encoder that processes 5 ms inputs; see also figure 1(a)).

The separator learns a mask to extract clean signals from noisy inputs. This is achieved using two spiking modules, SCNN and SRNN (figures 3(b) and 3(c), respectively). Initially, the 2D feature map generated by the encoder is subjected to a layer normalization, followed by a bottleneck $B \times 1$ convolution layer, resulting in an output with B channels. To reduce computations in the subsequent SCNN layer, the bottleneck convolution layer is followed by a binarization layer incorporating an activation suppression operation [22]. This binarization layer employs a learnable threshold to convert values below the threshold to 0 and those above it to 1. After processing through the SCNN and SRNN layers, the output is fed into a fully-connected readout layer using non-spiking ALIF neurons, as illustrated in figure 3(d). The readout layer's activations undergo suppression through a sparsification layer. Similar to the binarization layer preceding the SCNN layer, the sparsification layer uses a learnable threshold to set values below the threshold to 0. However,

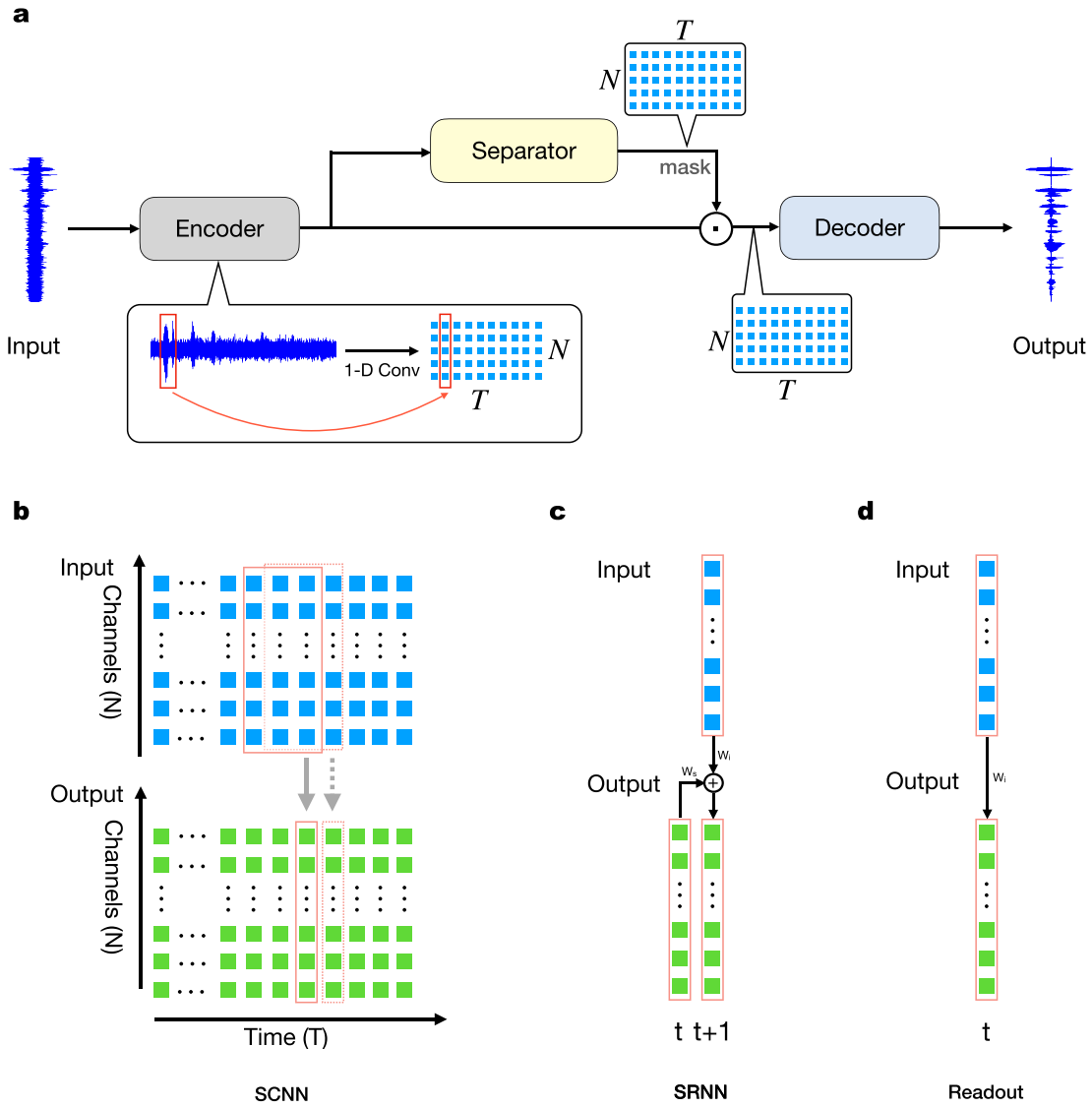

Figure 3. (a) In the mask-based encoder-separator-decoder architecture, the encoder converts overlapping frames into 1D features through convolution and aligns them into a 2D feature map. Each 1D feature is processed in one time step in the subsequent spiking layers. (b) In the SCNN layer, a group convolution is applied along the temporal axis of a feature map to capture temporal contextual information. (c) The SRNN layer is a fully-connected recurrent spiking layer that integrates contexts along the frequency direction of its input 2D feature map. (d) Readout is done using a fully-connected layer with non-spiking ALIF neurons, where the membrane potential of these neurons is calculated and output without any spiking or resetting.

unlike the binarization layer, it preserves values above the threshold without altering them. This suppressed output is then passed through a $N \times 1$ convolution layer with the Sigmoid activation function to produce a mask for clean signals. Finally, the 2D feature map output by the encoder, multiplied by the mask, is converted back to a 1D enhanced signal by the decoder, which comprises a deconvolutional layer.

The following provides more details on the SCNN and SRNN layers:

a) *SCNN layer.* The SCNN layer takes in the features output by the binarized bottleneck layer and integrates context along its temporal direction. As illustrated in figure 3(b), a group convolution is carried out along the temporal axis of the features (B channels) to integrate contexts of a predefined number of previous time steps, producing an output feature map with H channels. Such time steps are called *context steps*. In this layer, context from waveform signals can be integrated beyond the duration of the encoder's input frame. For one time step of this layer, each input channel is convolved with a set of $\frac{H}{B}$ filters; the input feature map is zero-padded before the first time step. PLIF neurons [36] are applied in this layer, with the surrogate gradient function $\sigma(x) = \frac{1}{\pi}\arctan(\pi x) + \frac{1}{2}$.

b) *SRNN layer.* The SRNN layer, based on the SRNN introduced in [21], is a fully-connected recurrent layer of spiking neurons that captures contexts along the frequency direction to extract frequency-related features. Taking the output from the SCNN layer, it produces outputs with B channels. We apply ALIF neurons [37] in this layer; for training, the multi-Gaussian surrogate gradient function [21] is used.

In this design, the SCNN layer captures contextual information across the temporal axis of the encoder's feature map, while the SRNN layer integrates contexts along the frequency axis. As we will demonstrate, together they form a robust separator capable of precise mask generation, particularly effective for handling extended audio sequences while incurring low-latency.

3.4. Training

The loss function used to train our model consists of three components. The first component ($L_{\text{si-snr}}$) maximizes SI-SNRs of enhanced waveforms. The second component (L_{mse}) is a Mean-Square-Error loss that minimizes the squared L_2 norm between enhanced waveforms and clean waveforms. The third component consists of two L_1 regularizers (L_1^{binary} and L_1^{sparse}) that penalize the non-zero activations output by the binarization layer and the sparsification layer, respectively [22]. Overall, the loss function is:

$$L = 100 + L_{\text{si-snr}} + 0.001 * L_{\text{mse}} + \lambda_1 * L_1^{\text{binary}} + \lambda_2 * L_1^{\text{sparse}}, \tag{10}$$

where $\lambda_1 = 0.001$ and $\lambda_2 = 0.001$ are optimized through grid search.

4. Experiments

In this section, we first briefly describe the VCTK corpus and the Intel N-DNS Challenge dataset. We employ two categories of metrics: speech objective metrics to evaluate speech performance, and power metrics to assess energy efficiency. We examine different model configurations, including variations in frame lengths in the encoder, and benchmark our models against baseline methods. Additionally, we conduct an ablation study to analyze the contributions of individual model components to overall performance.

4.1. Datasets

a) *VCTK corpus.* The VCTK corpus encompasses 10 hours of speech data, with most utterances lasting no more than 5 seconds, and some as brief as 2 to 3 seconds. The training set, which we downsampled from the original 48 kHz to 16 kHz, comprises 11,575 sentences. This training dataset involves 28 speakers (14 males and 14 females), all sharing the same English region accent, with each speaker contributing around 400 sentences. The training dataset includes a set of ten types of noises, two artificially generated noise (speech-shaped noise and babble) and eight real noise recordings sourced from the demand database [38]. Four signal-to-noise ratios (SNRs) are considered: 15 dB, 10 dB, 5 dB, and 0 dB, resulting in 40 distinct noise conditions. The testing set for VCTK consists of 827 sentences and includes two speakers (one male and one female). To simulate real-world conditions, five additional noises from the demand database, different from those in the training set, are included under four SNRs: 17.5 dB, 12.5 dB, 7.5 dB, and 2.5 dB. This results in a total of 20 unique noise conditions. For the VCTK corpus, we use the Spiking-UNet287 [21] as our baseline, as it is the only SNN model that has reported results on this dataset.

b) *Intel N-DNS challenge dataset.* The Intel N-DNS challenge dataset, derived from the Microsoft DNS challenge dataset [12], contains 500 hours of speech data distributed across both training and validation sets. Each set comprises 60,000 samples. The dataset incorporates a range of SNRs from 20 dB to -5 dB. Each utterance maintains a fixed duration of 30 seconds. The speech samples include five languages: English, German, French, Spanish, and Russian, and they may be combined within a single sample.

4.2. Evaluation metrics

To assess the performance of DPSNNs, we use two distinct categories of evaluation metrics: speech objective metrics and power metrics.

1) *Speech objective metrics.* For objective evaluation, we use SNR metrics, speech quality metrics, and speech intelligibility metrics, which are briefly described below.
   a) *SNR metric.* For the evaluation of SNRs, we use the Scale-Invariant Signal-to-Noise Ratio (SI-SNR) [39]. SI-SNR is specifically designed to be scale-invariant so that changes in the overall amplitude (volume) of the enhanced signal do not influence its measurement. It is defined as follows:

$$\text{SI-SNR} := 10\log_{10}\frac{||\mathbf{s}_{\text{target}}||_2}{||\mathbf{e}_{\text{noise}}||_2}, \tag{11}$$

   where $\mathbf{s}_{\text{target}} := \frac{\tilde{\mathbf{s}}\cdot\mathbf{s}}{||\mathbf{s}||_2^2}\mathbf{s}$ and $\mathbf{e}_{\text{noise}} := \tilde{\mathbf{s}} - \mathbf{s}_{\text{target}}$. For the Intel N-DNS challenge dataset, we also evaluate SI-SNRi, which represents the SI-SNR improvement relative to the noisy data.

Table 1. The effect of different configurations on model objective metrics, as evaluated using the VCTK corpus. The length of input examples is 1 second. The frame length (L) in the encoder is 80. The size of the context steps in SCNN is 4.

					DNSMOS				
N	B	H	SI-SNR [dB]	PESQ	*OVRL*	*SIG*	*BAK*	Params	Learning Rate
256	256	256	18.13	2.23	2.79	3.21	3.59	372 K	1×10^{-2}
512	128	512	18.26	2.30	2.79	3.21	3.61	317 K	1×10^{-2}
512	256	512	18.34	2.32	2.85	3.23	3.73	613 K	7.5×10^{-3}
512	512	512	18.48	2.36	2.86	3.24	3.72	1.4 M	7.5×10^{-3}

b) *Speech quality metrics.* The first speech quality metric we use is the perceptual estimation of speech quality (PESQ) [40]. PESQ evaluates speech quality by comparing the clean and enhanced speech signals, taking into account factors such as distortion, noise, and speech intelligibility. It ranges from −0.5 to 4.5, with higher scores indicating better perceived quality.
We also calculate Distributed Network Speech Mean Opinion Score (DNSMOS) [41] to evaluate speech quality. Mean Opinion Score (MOS) is a subjective metric that derives an average opinion score from human evaluations to assess perceived speech quality. MOS scores range from 1 to 5, where 1 signifies poor quality and 5 denotes excellent quality. DNSMOS employs deep learning models to predict subjective quality ratings based on features extracted from audio signals. DNSMOS provides three scores that assess speech quality (SIG), background noise quality (BAK), and overall audio quality (OVRL).

c) *Intelligibility metric.* The Short-Time Objective Intelligibility (STOI) metric [42] is a method used to assess the intelligibility of speech signals. STOI compares the short-time spectral envelopes of the clean and enhanced speech signals to estimate how intelligible the enhanced signal is compared to the original signal. The resulting value ranges from 0 to 1, with 1 indicating perfect intelligibility and 0 indicating no intelligibility.

2) *Power metrics.* We apply two power metrics, namely the power proxy and Power Delay Product (PDP) proxy, both introduced in the Intel N-DNS challenge [12]. The power proxy evaluates the effective number of synaptic operations a model performs per second:

$$P_{\text{proxy}} = \text{Effective SynOPS} = \text{SynOPS} + 10 \times \text{NeuronOPS}, \tag{12}$$

where SynOPS and NeuronOPS represent the average number of synaptic operations and neuron updates performed per second, respectively. The PDP proxy enables comparisons among solutions that address trade-offs between latency and power consumption, and is defined as follows:

$$\text{PDP}_{\text{proxy}} = P_{\text{proxy}} \times L, \tag{13}$$

where L denotes the algorithmic latency of a model in seconds. Effectively, $\text{PDP}_{\text{proxy}}$ expresses the power consumed per frame. Note that for frequency-domain DNN models, only the synaptic operations and neuron updates in the DNN itself are calculated in the power proxy, excluding the power consumption of the STFT and iSTFT. When comparing with these models, we also disregard the power consumption of the corresponding components in our DPSNNs, specifically the encoder and decoder.

4.3. Optimizing the network parameters on VCTK

First, we assess the performance of our model on the VCTK corpus with channel combinations N, B, and H and present the results in table 1. For the definitions of N, B, and H, please refer to the introduction of the architecture in section 3 and figure 2. From our evaluation, we draw the following conclusions:

(i) Encoder/decoder: expanding the number of channels enhances frequency resolution, thereby improving overall performance.

(ii) Channels in the separator: employing a small bottleneck channel B alongside a large number of channels H within the spiking block(s) proves effective. Additionally, larger B values consistently outperform smaller ones, a deviation from the findings in [7] where the optimal H/B value was found to be around 5. This discrepancy may be attributed to the nature of SNNs, which produce binary outputs and therefore require more neurons to convey information.

Examining our approach in detail, we find that the performance of our model is significantly influenced by the length of input examples, both for training and for evaluation. We explored our DPSNN models with

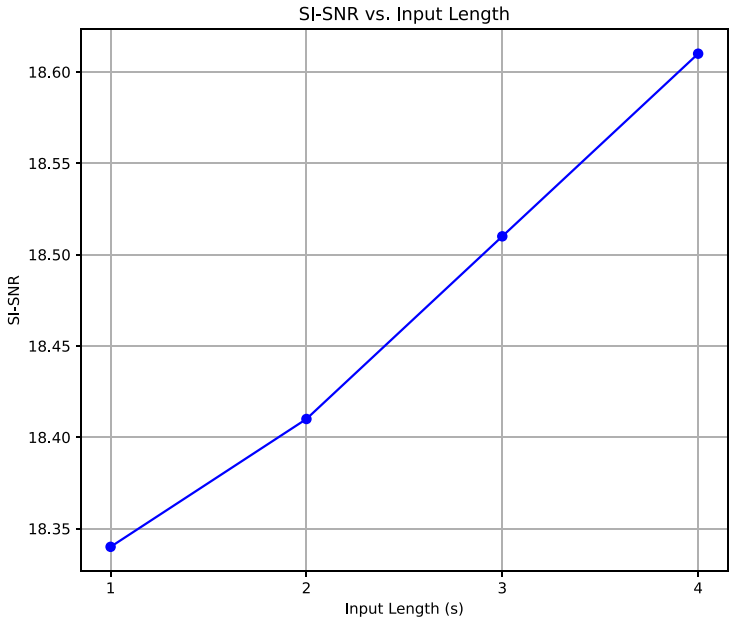

Figure 4. Influence of the length of input examples on model performance, as evaluated on the testing set of VCTK. The channels in the model are $N = 512, B = 256, H = 512$. The frame length ($L$) in the encoder is 80. The size of the context steps in SCNN is 4.

Table 2. Influence of the context steps in SCNN on the model objective performance, as evaluated using the VCTK corpus. The length of input examples is 1 second. The frame length (L) in the encoder is 80. The channels in the model are $N = 512, B = 256, H = 512$.

			DNSMOS		
Context Steps	SI-SNR [dB]	PESQ	*OVRL*	*SIG*	*BAK*
2	18.28	2.29	2.81	3.20	3.67
4	**18.34**	**2.32**	**2.85**	**3.23**	**3.73**
8	18.16	2.29	2.79	3.21	3.60

Table 3. Influences of frame lengths in the encoder (L) on model objective metrics, as evaluated using the VCTK corpus. The length of input examples is 1 second. The channels in the model are $N = 512, B = 256, H = 512$. The size of the context steps in SCNN is 4.

				DNSMOS		
L	Latency [ms]	SI-SNR [dB]	PESQ	*OVRL*	*SIG*	*BAK*
40	2.5	17.65	2.24	2.80	3.20	3.66
80	5	18.34	2.32	2.85	3.23	3.73
160	10	18.38	2.28	2.79	3.20	3.62

input lengths from 1.0 second to 4.0 seconds on VCTK. As illustrated in figure 4, the model performs better with longer inputs, achieving the best SI-SNR result with the 4.0-second example length.

We further assessed the impact of the context steps in the SCNN module on performance. As shown in table 2 , models with four context steps generally achieve better results than those with two or eight context steps, in terms of both SI-SNR and speech quality metrics.

Finally, we conducted experiments to assess models with varying frame lengths (L) in the encoder. As can be seen in table 3, models with larger L exhibit improved SI-SNRs. We attribute this enhancement to the larger context provided by a larger L for each time step within the separator. In addition, it appears that frame lengths have a limited impact on the PESQ and DNSMOS measures, unlike SI-SNR. Furthermore, a longer frame increases model latency, which is not desirable. Therefore, to strike a balance between speech performance and latency, we selected $L = 80$ for most of our experiments.

4.4. Comparison of DPSNN with previous methods

The comparison of the best performing DPSNN model with previous SNN models on VCTK is presented in table 4. First, we find that our model incurs significantly lower latency (5 ms) compared to both the SDNN baseline model [12] and Spiking-UNet [20]. Furthermore, our model outperforms in terms of both DNSMOS and STOI metrics, while its PESQ performance falls slightly short compared to Spiking-UNet.

Table 4. Objective metric comparisons on the VCTK corpus. The length of input examples is 4 seconds. The frame length (L) in the encoder is 80. The size of the context steps in SCNN is 4. The channels in the DPSNN are $N = 512, B = 256, H = 512$.

Model	Latency [ms]	SI-SNR [dB]	PESQ	DNSMOS			STOI
				OVRL	*SIG*	*BAK*	
Noisy	—	8.44	1.97	2.69	3.34	3.12	—
SDNN baseline [20]	32	—	2.00	2.44	3.05	3.09	0.91
Spiking-UNet [20]	32	—	**2.66**	2.81	3.13	**3.85**	0.92
DPSNN	**5**	18.62	2.37	**2.94**	**3.27**	3.84	**0.93**

Table 5. Ablation study. The length of input examples is 1 second. The channels in the model are $N = 512, B = 512, H = 512$. The frame length (L) in the encoder is 80. The size of the context steps in SCNN is 4.

Ablation	SI-SNR [dB]	PESQ	DNSMOS		
			OVRL	*SIG*	*BAK*
DPSNN	18.48	2.36	2.86	3.24	3.72
w/o SCNN	17.72	2.18	2.68	3.19	3.37
w/o SRNN	18.38	2.32	2.82	3.22	3.67

Table 6. Evaluation metrics comparisons on the Intel N-DNS Challenge dataset. The frame length (L) in the encoder of DPSNNs is 80. The channels in the DPSNNs are $N = 512, B = 512, H = 512$. The size of the context steps is 12. Note that the power metrics (power proxy and PDP proxy) and parameter sizes for the DPSNNs exclude the power consumption of the encoder and decoder—the inclusive numbers are noted in the parenthesis.

Model	Latency [ms]	SI-SNR [dB]	SI-SNR [dB]	DNSMOS			Power Proxy [M-Ops/s]	PDP Proxy [M-Ops]	Params [M]
				OVRL	*BAK*	*SIG*			
SDNN baseline [12]	32	12.5	4.88	2.71	3.21	3.46	14.52	0.46	0.465
Spiking-FullSubNet (Large) [30]	32	14.80	7.43	3.03	3.33	3.96	74.10	2.37	1.29
Spiking S4 [19]	32	14.58	7.21	2.85	3.21	3.74	—	—	0.53
Spiking-FullSubNet (Small) [30]	32	13.89	6.52	2.97	3.28	3.93	29.24	0.94	0.953
CTDNN [18]	32	13.52	6.59	2.97	3.32	3.86	61.37	0.49	0.90
DPSNN (4.0-second inputs)	5	14.54	7.18	2.88	3.27	3.72	175.2 (208.0)	0.88 (1.04)	1.32 (1.40)
DPSNN (5.0-second inputs)	5	14.60	7.23	2.89	3.29	3.71	180.4 (213.1)	0.90 (1.07)	1.32 (1.40)
DPSNN (5.0-second inputs)	10	14.70	7.34	2.90	3.27	3.77	87.47 (120.23)	0.87 (1.20)	1.32 (1.40)

This disparity may be attributed to the reliance of PESQ computation on the magnitude spectrogram of speech, as discussed in [8], where a similar observation was made.

4.5. Ablation study

We conducted experiments on VCTK to evaluate the impact of the SCNN and SRNN layers on our model's performance. We removed either the SCNN or SRNN layer, and the results are presented in table 5. We find substantial decline in overall performance in either case, in particular when removing the SCNN module.

4.6. Intel N-DNS challenge dataset

We evaluate the DPSNNs against several benchmarks using the Intel N-DNS challenge dataset (table 6). These benchmarks include the SDNN baseline model [12], the winning models from the Intel N-DNS challenge (Spiking-FullSubNet (Large) and Spiking-FullSubNet (Small) [30]), the runner-up model (CTDNN), and the Spiking S4 model [19].

In terms of speech objective metrics, our best SI-SNR and SI-SNRi values exceed those of Spiking S4, Spiking-FullSubNet (Small), and CTDNN. Additionally, our DNSMOS performance outperforms that of Spiking S4. Although our speech objective performance is slightly behind that of Spiking-FullSubNet (Large), our 5 ms latency substantially lower than these benchmark models, which all exhibit a latency of 32 ms.

As mentioned above, simply reducing the frame length of an existing frequency-domain speech enhancement model to lower latency results in decreased speech performance. We evaluated the performance of the SDNN baseline model [12] while altering its input frame lengths. The model was trained

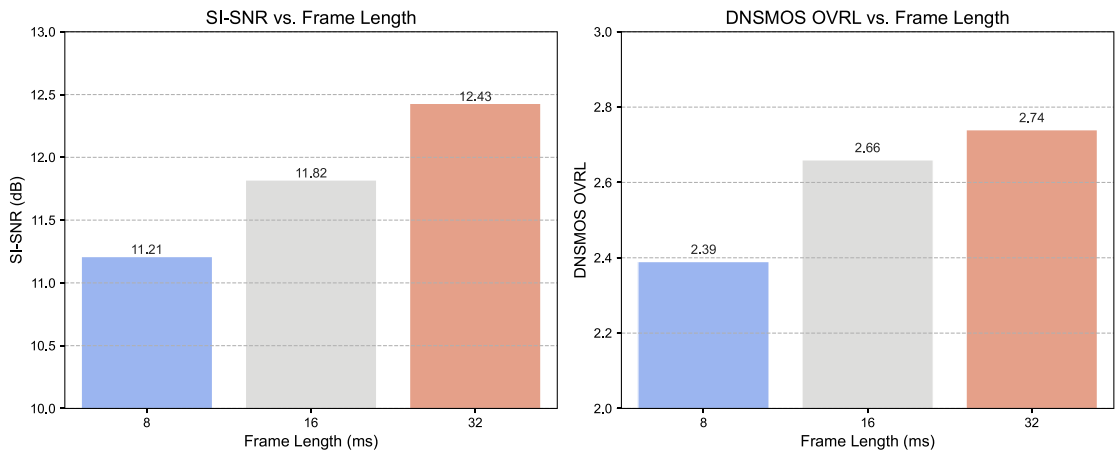

Figure 5. Performance of the SDNN baseline model [12] across varying frame lengths. The model was trained using different frame lengths (8 ms, 16 ms, and 32 ms) and then evaluated on the Intel N-DNS Challenge dataset. We observe that the frame length significantly influences both SI-SNR and DNSMOS OVRL.

with different frame lengths (8 ms, 16 ms, and 32 ms) and subsequently tested on the Intel N-DNS Challenge dataset. The results show a significant impact of frame length on both the SI-SNR and DNSMOS OVRL (figure 5): while the longer frame length (32 ms) yields the highest SI-SNR, shorter frame lengths (8 ms and 16 ms) result in a noticeable decline in both metrics. Additionally, increasing the frame length from 5 ms to 10 ms for the DPSNN improves its speech performance slightly while halving the power proxy (table 6). These results align with the findings in [17], which reported a degradation in Signal-to-Distortion Ratio (SDR) when reducing the frame length.

Compared to the other models in table 6, our DPSNNs show a relatively higher power proxy values, which indicates that DPSNNs perform more synaptic operations over the same audio duration. However, DPSNNs perform fewer operations per time step, resulting in greater energy efficiency for each time step compared to both the Spiking-FullSubNet (Large) and CTDNN. We simply require more time steps due to the smaller input frame size, which is the trade-off we must accept to achieve low latency. If these other solutions were implemented with reduced latency, they would incur considerably higher power consumption than DPSNNs operating at the same latency. This is demonstrated by the PDP proxy, a joint evaluation metric for latency and power efficiency, which shows that our DPSNN models significantly outperform both leading Spiking-FullSubNet models. Although the SDNN baseline and CTDNN have lower values in this specific metric, their speech metrics (SI-SNR and/or DNSMOS) fall well short of those achieved by our DPSNN models. This highlights the combined capability of our DPSNNs in terms of speech quality, latency, and power consumption.

5. Conclusion

Speech enhancement is crucial for improving spoken communication in noisy environments, with applications spanning ASR, hearing aids, and telecommunications. Recognizing the challenges posed by power constraints and the need for low latency in these domains, we address the current limitations of effective frequency-domain SNN solutions, which typically impose substantial latency due to long temporal windows used for their STFTs and iSTFTs to ensure satisfactory enhancement performance.

Drawing inspiration from the efficacy of DPRNNs in deep learning, we introduce the time-domain DPSNN as a novel two-phase SNN framework for low-latency speech enhancement. In the DPSNN framework, the first phase uses SCNNs to capture temporal contextual information, while the second phase employs SRNNs to focus on frequency-related features. The DPSNN relies on a learned encoder to encode input waveforms and a decoder to revert the encoder outputs back to the sound waveforms. This encoder-decoder configuration enables the processing of short input frames, resulting in low latency in time-domain models. In addition, the threshold-based activation suppression combined with L_1 regularization loss are applied to specific non-spiking layers to further enhance energy efficiency of DPSNNs.

Evaluating on benchmark datasets such as VCTK and Intel N-DNS Challenge dataset, our approach achieves significantly lower latency, approximately 5 ms, compared to current solutions, while maintaining excellent SNR, speech quality, intelligibility, and energy efficiency. Notably, the strong performance of DPSNNs in speech metrics suggests that the filters learned by the encoders of DPSNNs are optimized for low

frequencies crucial for speech intelligibility, similar to those in ConvTasNet [8]. Furthermore, the SCNN layer allows the integration of context from previous time steps to capture long-range information in the waveform signals. This tuning enables DPSNNs to achieve low latency while preserving speech performance. Overall, DPSNNs represent a significant advancement in speech enhancement techniques, offering improved communication experiences and energy efficiency across various applications.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://datashare.ed.ac.uk/handle/10283/2791 and https://github.com/IntelLabs/IntelNeuromorphicDNSChallenge#dataset.

Acknowledgments

This work was inspired by the discussions in the The CapoCaccia Workshops toward Neuromorphic Intelligence 2023. The authors also express their appreciation to Dr. Bojian Yin for providing valuable insights concerning SNN training, to Dr. Guangzhi Tang and Dr. Paul Detterer for their constructive suggestions on model sparsity, and to Haohui Zhang for his valuable help regarding the figures. TS is supported by NWO-NWA Grant NWA.1292.19.298. SB is supported by the European Union (Grant Agreement 7202070 'HBP').

ORCID iDs

Tao Sun https://orcid.org/0000-0002-8967-8760
Sander Bohté https://orcid.org/0000-0002-7866-278X

References

[1] Wang D and Chen J 2018 Supervised speech separation based on deep learning: an overview *IEEE/ACM Trans. Audio, Speech Lang. Process.* **26** 1702–26
[2] Wang Y and Wang D 2013 Towards scaling up classification-based speech separation *IEEE Trans. Audio, Speech Lang. Process.* **21** 1381–90
[3] Xu Y, Du J, Dai Li-R and Lee C-H 2014 A regression approach to speech enhancement based on deep neural networks *IEEE/ACM Trans. Audio, Speech Lang. Process.* **23** 7–19
[4] Pandey A and Wang D 2018 A new framework for supervised speech enhancement in the time domain *Proc. Interspeech 2018* pp 1136–40
[5] Gerkmann T, Krawczyk-Becker M and Le Roux J 2015 Phase processing for single-channel speech enhancement: History and recent advances *IEEE Signal Process. Mag.* **32** 55–66
[6] Fu S-W, Hu T-yao, Tsao Y and Lu X 2017 Complex spectrogram enhancement by convolutional neural network with multi-metrics learning *2017 IEEE 27th Int. Workshop on Machine Learning for Signal Processing (MLSP)* (IEEE) pp 1–6
[7] Luo Y and Mesgarani N 2018 TasNet: time-domain audio separation network for real-time, single-channel speech separation *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (IEEE) pp 696–700
[8] Luo Y and Mesgarani N 2019 Conv-TasNet: surpassing ideal time-frequency magnitude masking for speech separation *IEEE/ACM Trans. Audio, Speech Lang. Process.* **27** 1256–66
[9] Luo Y, Chen Z and Yoshioka T 2020 Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation *ICASSP 2020-2020 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE) pp 46–50
[10] Drgas S 2023 A survey on low-latency DNN-based speech enhancement *Sensors* **23** 1380
[11] Wang Z-Q, Wichern G, Watanabe S and Le Roux J 2022 STFT-domain neural speech enhancement with very low algorithmic latency *IEEE/ACM Trans. Audio, Speech Lang. Process.* **31** 397–410
[12] Timcheck J, Bam Shrestha S, Ben Dayan Rubin D, Kupryjanow A, Orchard G, Pindor L, Shea T and Davies M 2023 The Intel neuromorphic DNS challenge *Neuromorph. Comput. Eng.* **3** 034005
[13] Wang S, Naithani G, Politis A and Virtanen T 2021 Deep neural network based low-latency speech separation with asymmetric analysis-synthesis window pair *2021 29th European Signal Processing Conf. (EUSIPCO)* (IEEE) pp 301–5
[14] Koizumi Y, Harada N and Haneda Y 2019 Trainable adaptive window switching for speech enhancement *ICASSP 2019-2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE) pp 616–20
[15] ITU T 2000 Recommendation g. 114, one-way transmission time *Series G: Transmission Systems and Media, Digital Systems and Networks, Telecommunication Standardization Sector of ITU*
[16] Team Clarity 2024 Clarity challenge: speech enhancement for hearing aids (available at: https://claritychallenge.org/ICASSP2023_announcement_page#page-top) (Accessed 08 October 2024)
[17] Wang S, Naithani G and Virtanen T 2019 Low-latency deep clustering for speech separation *ICASSP 2019-2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE) pp 76–80
[18] The Intel neuromorphic DNS challenge 2024 (available at: https://github.com/IntelLabs/IntelNeuromorphicDNSChallenge#metricsboard) (Accessed 16 April 2024)
[19] Du Y, Liu X and Chua Y 2023 Spiking structured state space model for monaural speech enhancement (arXiv:2309.03641)
[20] Riahi A and Plourde Eric 2023 Single channel speech enhancement using U-Net spiking neural networks *2023 IEEE Canadian Conf. on Electrical and Computer Engineering (CCECE)* (IEEE) pp 111–6

[21] Yin B, Corradi F and Bohté S M 2021 Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks *Nat. Mach. Intell.* **3** 905–13
[22] Zhu Z, Pourtaherian A, Waeijen L, Bondarev E and Moreira O 2023 STAR: sparse thresholded activation under partial-regularization for activation sparsity exploration *2023 IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)* (IEEE) pp 4554–63
[23] Valentini-Botinhao C *et al* 2017 Noisy speech database for training speech enhancement algorithms and tts models *University of Edinburgh. School of Informatics. Centre for Speech Technology Research (Cstr)*
[24] Sun T 2022 *Time-Domain Deep Neural Networks for Speech Separation* (Ohio University)
[25] Pandey A and Wang D 2019 TCNN: temporal convolutional neural network for real-time speech enhancement in the time domain *2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (IEEE) pp 6875–9
[26] Fu S-W, Tsao Y, Lu X and Kawai H 2017 Raw waveform-based speech enhancement by fully convolutional networks *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conf.* (IEEE) pp 006–12
[27] Gong S, Wang Z, Sun T, Zhang Y, Smith C D, Xu Li and Liu J 2019 Dilated FCN: listening longer to hear better *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (IEEE) pp 254–8
[28] Macartney C and Weyde T 2018 Improved speech enhancement with the Wave-U-Net (arXiv:1811.11307)
[29] Stoller D, Ewert S and Dixon S 2018 Wave-U-Net: a multi-scale neural network for end-to-end audio source separation (arXiv:1806.03185)
[30] Hao X, Ma C, Yang Q, Chen Tan K and Wu. J 2024 When audio denoising meets spiking neural network *2024 IEEE Conf. on Artificial Intelligence (CAI)* (IEEE) pp 1524–7
[31] Gu A, Goel K and Ré C 2022 Efficiently modeling long sequences with structured state spaces *Int. Conf. on Learning Representations (ICLR)*
[32] Ronneberger O *et al* 2015 U-Net: convolutional networks for biomedical image segmentation *Medical Image Computing and Computer-Assisted Intervention–Miccai 2015* (Springer) pp 234–41
[33] Sun T, Gong S, Wang Z, Smith C D, Wang X, Xu Li and Liu J 2021 Boosting the intelligibility of waveform speech enhancement networks through self-supervised representations *2021 20th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)* (IEEE) pp 992–7
[34] Gerstner W and Kistler W M 2002 *Spiking Neuron Models: Single Neurons, Populations, Plasticity* (Cambridge University Press)
[35] Neftci E O, Mostafa H and Zenke F 2019 Surrogate gradient learning in spiking neural networks: bringing the power of gradient-based optimization to spiking neural networks *IEEE Signal Process. Mag.* **36** 51–63
[36] Fang W, Yu Z, Chen Y, Masquelier T, Huang T and Tian Y 2021 Incorporating learnable membrane time constant to enhance learning of spiking neural networks *CVPR* pp 2661–71
[37] Yin B, Corradi F and Bohté S M 2020 Effective and efficient computation with multiple-timescale spiking recurrent neural networks *Int. Conf. on Neuromorphic Systems 2020* pp 1–8
[38] Thiemann J, Ito N and Vincent E 2013 The diverse environments multi-channel acoustic noise database (DEMAND): a database of multichannel environmental noise recordings *Proc. Meetings on Acoustics* vol 19 (AIP Publishing)
[39] Le Roux J, Wisdom S, Erdogan H and Hershey J R 2019 SDR - half-baked or well done? *ICASSP 2019-2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE) pp 626–30
[40] Rix A W, Beerends J G, Hollier M P and Hekstra A P 2001 Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs *2001 IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (*Proc. (Cat. No. 01CH37221)*) vol 2 (IEEE) pp 749–52
[41] Reddy C K A, Gopal V and Cutler R 2022 DNSMOS P. 835: a non-intrusive perceptual objective speech quality metric to evaluate noise suppressors *ICASSP 2022-2022 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE) pp 886–90
[42] Taal C H, Hendriks R C, Heusdens R and Jensen J 2011 An algorithm for intelligibility prediction of time–frequency weighted noisy speech *IEEE Trans. Audio, Speech Lang Process.* **19** 2125–36