

Acoustic neural networks: Identifying design principles and exploring physical feasibility

Ivan Kalthoff,^{1, 2, 3} Marcel Rey,⁴ and Raphael Wittkowski^{1, 2, 3, *}

¹*Department of Physics, RWTH Aachen University, 52074 Aachen, Germany*

²*DWI – Leibniz Institute for Interactive Materials, 52074 Aachen, Germany*

³*Institute of Theoretical Physics, Center for Soft Nanoscience,
University of Münster, 48149 Münster, Germany*

⁴*Institute of Physical Chemistry, Center for Soft Nanoscience,
University of Münster, 48149 Münster, Germany*

Wave-guide-based physical systems provide a promising route toward energy-efficient analog computing beyond traditional electronics. Within this landscape, acoustic neural networks represent a promising approach for achieving low-power computation in environments where electronics are inefficient or limited, yet their systematic design has remained largely unexplored. Here we introduce a framework for designing and simulating acoustic neural networks, which perform computation through the propagation of sound waves. Using a digital-twin approach, we train conventional neural network architectures under physically motivated constraints including non-negative signals and weights, the absence of bias terms, and nonlinearities compatible with intensity-based, non-negative acoustic signals. Our work provides a general framework for acoustic neural networks that connects learnable network components directly to physically measurable acoustic properties, enabling the systematic design of realizable acoustic computing systems. We demonstrate that constrained recurrent and hierarchical architectures can perform accurate speech classification, and we propose the SincHSRNN, a hybrid model that combines learnable acoustic bandpass filters with hierarchical temporal processing. The SincHSRNN achieves up to 95% accuracy on the AudioMNIST dataset while remaining compatible with passive acoustic components. Beyond computational performance, the learned parameters correspond to measurable material and geometric properties such as attenuation and transmission. Our results establish general design principles for physically realizable acoustic neural networks and outline a pathway toward low-power, wave-based neural computing.

I. INTRODUCTION

Deep learning has transformed fields such as speech recognition, image classification, and scientific data analysis.¹ Its rapid progress has been enabled primarily by advances in hardware rather than fundamentally new algorithms.² However, sustaining this trajectory is increasingly difficult as transistor miniaturization reaches physical limits.³ This has motivated research into alternative paradigms such as neuromorphic, quantum, and wave-based computing.^{2,4}

In this context, *acoustic neural networks*, architectures that process information via the propagation of sound waves rather than electronic signals, represent an emerging yet largely unexplored direction. Prior work in acoustic metamaterials has demonstrated that engineered wave structures can perform analog mathematical operations such as differentiation, equation solving, and edge detection, and even enable neural-like inference through passive meta-neural and mechanical neural networks.^{5–10} Together, these results demonstrate that wave-based physical systems can implement fundamental operations similar to those used in neural networks. In acoustic neural networks, the wave amplitude or intensity carries the signal, while material and geometric properties determine how signals combine and attenuate. These amplitude

modulations act as the physical counterpart of neural network weights, setting the relative influence of different inputs. Similar to optical neural networks,^{11,12} acoustic neural networks can perform computations directly on wave signals without analog-to-digital conversion, reducing power consumption and latency.^{5,10,13,14} A key challenge remains the realization of nonlinear activation functions, which are essential for neural computation because they allow networks to represent complex, nonlinear relationships that cannot be captured by linear wave propagation alone. While nonlinear effects can also occur in conventional materials at high acoustic intensities, when the regime of linear elasticity is exceeded, their magnitude and tunability are generally limited. Recent advances in nonlinear acoustic materials and metamaterials, however, demonstrate stronger and design-controlled nonlinear effects, specifically in the form of amplitude-dependent attenuation, where the acoustic loss can increase, decrease, or saturate with growing intensity,^{15–18} providing a promising physical mechanism for such nonlinear transformations.

Acoustic neural networks could open the door to low-power applications ranging from direct analog speech recognition and smart hearing aids to computing in environments where conventional electronics are limited.^{19,20} In the context of hearing aids, such systems could help identify relevant acoustic patterns, such as warning signals, while operating with minimal energy consumption. The demand for low-power, edge-based processing of

* Corresponding author: rgwitt25@dwi.rwth-aachen.de

acoustic information further motivates the design of neural architectures composed of acoustic elements.¹⁴

Wave-based physical neural networks, including optical systems, acoustic and mechanical metamaterials, and analog recurrent structures, have demonstrated that wave propagation can perform learned transformations in hardware. However, within the acoustic domain in particular, existing implementations remain fixed-function or limited-operation systems, typically realizing a single trained mapping or isolated operations such as filtering, differentiation, or inference.^{5–8} Despite partial reconfigurability or in-situ training in some systems, there is still no general framework for constructing full neural architectures whose parameters map directly onto physically realizable acoustic processes. Here, we address this gap by introducing a unified digital-twin approach that constrains all computations to passive, non-negative acoustic operations and whose parameters map directly onto measurable transmission, attenuation, and nonlinear loss. This perspective treats acoustic neural networks not as single engineered devices but as broadly applicable computational architectures grounded in the physics of sound. In turn, this framework enables the systematic design and optimization of future low-power acoustic neural hardware directly informed by the learned structure of the digital twin.

To realize this framework in practice, we develop constrained digital models that serve as blueprints for analog acoustic neural networks. Our models restrict signals and weights to non-negative values (representing sound intensity and attenuation), omit bias terms, and employ activation functions compatible with non-negative signals. The inherently temporal nature of acoustic signals, together with the cumulative attenuation that arises during wave propagation, makes recurrent neural networks (RNNs) the simplest computational architecture that mirrors the dynamical behavior of physical acoustic systems. More expressive gated models such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) networks^{21,22} rely on multiplicative gating mechanisms and signed activations that are incompatible with passive, non-negative acoustic media, making simple RNNs the physically realizable choice. Likewise, we evaluate all architectures directly on raw audio intensities, which correspond to the physical input available in an acoustic system and avoid digital preprocessing steps such as spectrogram computation that would break the analog computing paradigm.

Building on this conceptual framework, we introduce a set of constrained architectures for speech classification on the AudioMNIST dataset.²³ Starting from RNNs that capture temporal structure, we introduce hierarchical subsampling to handle long sequences, and finally propose a Hierarchical Subsampling Recurrent Neural Network with learnable sinc-based bandpass filters at the input stage (referred to as SincHSRNN). These sinc filters provide an interpretable and physically realizable mechanism for frequency-selective preprocessing, while

the hierarchical recurrent component captures temporal dependencies. The SincHSRNN achieves competitive performance while remaining compatible with acoustic constraints.

The article is organized as follows: Sec. II introduces the underlying concept of acoustic neural networks, explains how neural operations can be mapped onto physically realizable acoustic components, and describes the digital-twin framework, network architectures, dataset and training procedures used to implement and evaluate the constrained models. In Sec. III, we present and discuss the results for recurrent, hierarchical, and sinc-based architectures, highlighting the influence of acoustic constraints on model performance and stability. Finally, Sec. IV summarizes the key findings and outlines design principles for future physically realizable acoustic neural networks.

II. METHODS

A. Mapping acoustic neural networks onto physical systems

Acoustic neural networks aim to perform neural network computation through the physical propagation of sound waves, analogous to optical neural networks that use light for information processing.^{11,12,24} In such systems, the amplitude or intensity of an acoustic wave represents the signal transmitted between neurons, while the physical properties of the propagation medium determine the connection strength. This approach offers a path toward low-power, analog computing architectures that directly operate on acoustic signals with minimal digital processing.

1. Mapping neural operations to acoustic processes

The conceptual structure of an acoustic neural network is illustrated in Fig. 1. The functional elements of conventional neural networks can be mapped to acoustic processes. A connection weight corresponds to the transmission coefficient of an acoustic path, which depends on material attenuation, geometry, and reflection at interfaces. As acoustic systems are inherently passive, no amplification occurs, constraining weights to the range $w \in [0, 1]$. The system considered here includes no external power source, and summation of incoming signals arises naturally from the superposition of converging waves. A bias term could, in principle, be implemented as a constant external sound source, but its realization would require an additional emitter per neuron and is therefore omitted for practical simplicity.

Nonlinear activation functions can be introduced by exploiting intensity-dependent attenuation effects in nonlinear acoustic media. These mechanisms produce output amplitudes that vary nonlinearly with input intensity,

analogous to activation functions in digital networks. In this way, a material with intensity-dependent transmission effectively acts as an acoustic neuron. To illustrate the plausibility of such nonlinear behavior in acoustic systems, an offset version of the rectified linear unit (ReLU)

$$f(I) = \max(I - I_c, 0) \quad (1)$$

can serve as an example of an activation suitable for non-negative neural networks,²⁵ where I denotes the input intensity and I_c a characteristic threshold. This function captures the behavior of an acoustic element whose transmission remains negligible for intensities below I_c and increases linearly once the threshold is exceeded. Such amplitude-gated transmission has been observed in nonlinear acoustic materials and metamaterials.^{17,18}

2. Digital twin framework

To explore this concept systematically, the networks studied in this work are designed as digital twins of physically realizable acoustic systems. Each digital parameter corresponds directly to an acoustic property: non-negative input signals represent sound intensity, weights are restricted to $[0, 1]$ to model transmission losses, and nonlinearities are limited to functions compatible with non-negative signals realizable in known acoustic materials. Bias terms are excluded, and summation is implemented through linear superposition. It should be noted that the internal signal processing within the network is designed to be acoustically realizable. The output of the network, however, can be handled in different ways: one option is to process the final neuron signals digitally, for example by applying a softmax function to obtain class probabilities, while another is to use simple electronic components, such as comparators or multiplexers, to detect the strongest output signal and trigger a corresponding response. In both cases, the computationally demanding signal processing remains acoustic, whereas only the output stage involves minimal electronic operation.

The digital twin thus provides a simulation environment for identifying suitable architectures and parameter regimes. After training, the resulting network weights and structure define a blueprint for a potential acoustic implementation, guiding the selection of material properties, geometries, and nonlinear elements.

B. Network architectures

In choosing the model architectures, we restrict ourselves to recurrent structures whose computations can be mapped onto passive acoustic processes. Simple RNNs are compatible with these constraints because their recurrence relation requires only linear superposition, attenuation, and a scalar nonlinearity, all operations that can

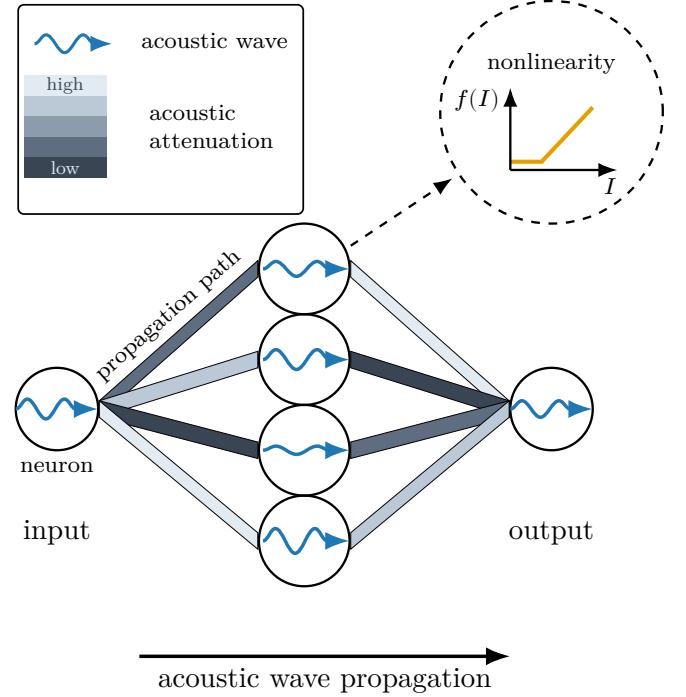


FIG. 1. Conceptual schematic of an acoustic neural network. Information is transmitted and processed using sound waves instead of electrical signals. Propagation paths act as attenuating connections with transmission coefficients corresponding to neural weights ($w \in [0, 1]$). Nonlinear transformations arise from intensity-dependent attenuation within acoustic media, serving as the activation function. Summation of signals occurs through the superposition of converging waves.

be realized through acoustic transmission and intensity-dependent attenuation. Likewise, we design our models to operate directly on the non-negative acoustic intensity values that would be available in a physical system, rather than on digitally preprocessed representations such as spectrograms or Mel-frequency cepstral coefficients (MFCCs). Avoiding such preprocessing ensures that all computations within the model correspond to transformations that could, in principle, be realized acoustically. Three classes of neural network architectures were considered, each reflecting an increasing level of sophistication. In this work, model capacity refers to the number of trainable parameters determined by the network architecture, which increases with the number of hidden units and layers.

1. Recurrent neural networks (RNNs)

As a baseline for sequential modeling, simple RNNs were trained on the binary subset of AudioMNIST (digits 0 and 1). In constrained variants, weights were clamped to the range $[0, 1]$ to model acoustic attenuation, bias terms were removed, and activation functions suitable for non-negative signals were employed, such as an offset ver-

sion of the absolute value or ReLU function.²⁵ These restrictions emulate the physical constraints of real acoustic media but limit model expressiveness and stability for long input sequences. Each RNN consists of a single recurrent layer with 8 to 64 hidden units, followed by a fully connected output layer that maps the final hidden state to the class logits for the softmax classifier. The unconstrained model uses a standard tanh activation, whereas the constrained model employs the offset non-negative activations mentioned above. All weights are initialized from a uniform distribution, and hidden states are initialized with small random values in the same range. Detailed training parameters and hyperparameter settings for the RNN models are summarized in Appendix A.

2. Hierarchical subsampling recurrent neural networks (HSRNNs)

To address the difficulty of modeling long raw audio sequences, HSRNNs²⁶ were employed. These architectures reduce the temporal resolution between recurrent layers through hierarchical subsampling, effectively shortening sequence length while preserving relevant information. Each HSRNN consists of multiple recurrent layers with increasing hidden dimensions and a subsampling step between layers. For each layer, the input sequence is divided into non-overlapping segments defined by a predefined subsampling factor. Each element within a segment is processed by its own position-specific weight matrix, and the resulting outputs are summed and combined with the recurrent input from the previous hidden state. The unconstrained model uses the standard tanh activation, while the constrained model replaces it with an offset absolute or offset ReLU activation to maintain non-negative hidden states. The output sequence of one layer serves as the input to the next. After the final recurrent layer, a fully connected layer with tanh activation precedes a linear classifier that maps to the class logits for the softmax. Bias terms are disabled in the constrained model, and weights are initialized using Xavier-uniform initialization, with constrained variants enforcing non-negative weights through initialization and clamping. The complete set of training parameters and subsampling configurations used for the HSRNNs is provided in Appendix A.

3. Hierarchical subsampling recurrent neural networks with learnable sinc filters (SincHSRNNs)

Our main contribution is the SincHSRNN, which combines learnable sinc filters²⁷ at the input stage with the hierarchical recurrent structure. The overall architecture of the proposed model is illustrated in Fig. 2. The sinc filters act as parameterized bandpass filters, initialized with lower and upper cut-off frequencies equally spaced on the Mel scale and windowed with a Hamming window. Since

their operation directly corresponds to acoustic bandpass filtering, these filters provide an interpretable and physically realizable front-end that mimics acoustic filtering. The filtered outputs are passed to the HSRNN, which captures temporal dependencies at multiple resolutions. The sinc filter layer was configured with a kernel size of 101 samples and five output channels. The kernel size defines the temporal window and therefore the frequency resolution of each bandpass filter, while the number of output channels determines how many parallel filters or learnable frequency bands are applied. The filtered outputs are passed to the HSRNN backbone, which consists of three to four recurrent layers with increasing hidden dimensions and a subsampling factor of eight between layers. The final recurrent output is processed by two fully connected layers with tanh activations, followed by a linear output layer and a softmax classifier producing the digit-class probabilities. In the constrained variant, the same structure is used but with non-negative weights, no bias terms, and non-negative activations such as the offset absolute value function. Detailed hyperparameter values and training settings for the SincHSRNNs are listed in Appendix A.

C. Dataset and preprocessing

All evaluations were performed on the AudioMNIST dataset,²³ which contains 30 000 one-second recordings of spoken digits (0-9) from 60 speakers of different genders and ages. The dataset was divided into a train set (80%) and a test set (20%), ensuring that no speaker appears in both sets. This prevents speaker-specific overfitting and provides a realistic benchmark for digit recognition from raw audio.

Prior to training, all audio data underwent several pre-processing steps. Recordings were resampled to lower rates between 1-8 kHz to reduce sequence length. Each waveform was zero-padded to a fixed duration of one second to ensure consistent input size. Amplitudes were normalized to the range $[-1, 1]$, and subsequently squared to represent acoustic intensity. This final transformation aligns the digital representation with the physical interpretation of acoustic intensity, ensuring non-negative inputs consistent with the constraints of acoustic neural networks.

D. Training procedure

The models were trained using the Adam optimizer²⁸ with cross-entropy loss and a mini-batch size of 64. Adam was selected as it is a widely adopted optimizer that provides stable and efficient convergence in neural network training. Several learning rates were evaluated to ensure stable convergence, leading to the choice of an initial rate of 10^{-3} for training and a reduced rate of 10^{-4} for fine-tuning during the final epochs. In constrained

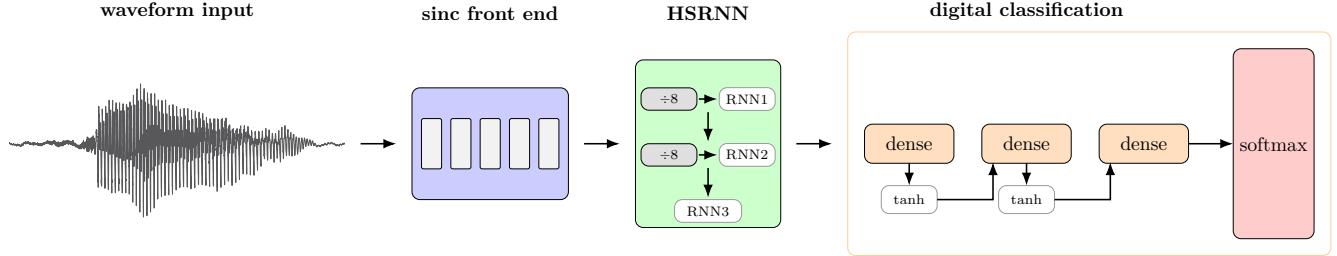


FIG. 2. Raw waveform input is processed by a front end of five learnable sinc filters that act as interpretable bandpass elements. The filtered signals are passed to a HSRNN composed of three to four recurrent layers, labeled RNN1–RNN3 in the figure, each preceded by temporal downsampling by a factor of 8 with learnable weights. The resulting feature representation is fed to two fully connected layers, denoted Dense in the figure, with tanh activations and a final fully connected output layer followed by a softmax classifier that produces the digit-class probabilities.

networks, all weights were clamped to $[0, 1]$ after each update, bias terms were omitted, and only non-negative activation functions (offset ReLU or absolute value) were used. Gradient clipping with a maximum norm of 1.0 stabilized training for recurrent models.

A crucial aspect for achieving convergence in the constrained networks was the choice of the weight initialization and activation function offset. Because all weights are restricted to positive values, conventional symmetric initializations (e.g., Xavier or Kaiming) caused many parameters to be clamped to zero, leading to near-constant activations and failed training. To address this, weights were initialized from a non-negative uniform distribution $\mathcal{U}(0, c)$, where the upper bound c was empirically tuned prior to training by analyzing the output activations of an untrained model. The offset parameter in the activation functions was adjusted in conjunction with c to keep activations within a range that avoids both vanishing and saturated regimes.

E. Evaluation metrics

The model performance was primarily evaluated using classification accuracy on the test set. To assess the robustness of the results and estimate statistical uncertainty, each model configuration was trained and evaluated five times with different random initializations. Reported accuracies and standard deviations are computed from these five independent runs.

III. RESULTS AND DISCUSSION

A. RNNs

To establish a baseline for speech classification on raw audio data, simple RNNs were trained on the binary subset of AudioMNIST (digits 0 and 1). These models capture temporal dependencies with minimal architectural

complexity, making them a suitable test case for assessing the feasibility of acoustic constraints.

Table I summarizes the train and test accuracies of unconstrained and constrained RNNs across different hidden-unit configurations at a fixed sampling rate of 1 kHz. For small networks with 8 or 16 hidden units, both model types achieve comparable test performance of approximately 68%, with differences well within the run-to-run variability. This shows that the temporal modeling capability of an RNN is largely retained even when constrained to non-negative weights and activations and trained without bias terms. Train and test accuracies remain closely aligned, suggesting minimal overfitting.

As model capacity increases, the constrained networks exhibit a pronounced decline in mean test accuracy and a substantial increase in variability across runs. In particular, architectures with 32 and 64 hidden units show sensitivity to initialization, with some models failing to train and remaining near random-guess performance ($\approx 50\%$) throughout all epochs. This instability reflects the limited flexibility introduced by non-negative weight constraints.

These findings align with previous work on non-negative neural networks,^{25,29,30} which reports that such constraints often lead to non-convergent or unstable training unless initialization schemes are adapted to compensate for the restricted parameter space. Because all weights are positive, conventional symmetric initializations result in many parameters being clamped to zero, yielding near-constant activations. This sensitivity underscores the importance of tailored initialization strategies for physically constrained architectures.

With suitable initialization, however, the constrained network achieves performance comparable to the unconstrained model, demonstrating that a physically plausible acoustic RNN can perform binary speech classification directly from raw waveform input. This is encouraging for real-world implementations, where simplicity and physical feasibility are crucial. Nevertheless, it is important to note that these results are obtained on a simplified binary classification task, where the overall accuracy

Hidden units	Unconstrained		Constrained	
	Train accuracy (%)	Test accuracy (%)	Train accuracy (%)	Test accuracy (%)
8	70.48 ± 0.08	68.58 ± 0.84	70.10 ± 0.51	68.03 ± 0.94
16	70.43 ± 0.18	68.02 ± 0.46	70.17 ± 0.37	68.66 ± 0.90
32	68.65 ± 3.17	68.21 ± 1.19	61.35 ± 9.36	60.64 ± 8.79
64	68.15 ± 3.20	66.98 ± 2.51	63.31 ± 8.17	62.66 ± 7.99

TABLE I. Mean train and test accuracies (%) with standard deviations for unconstrained and physically constrained RNNs on the binary AudioMNIST dataset at a sampling rate of 1 kHz. While both models achieve comparable performance for smaller architectures, the constrained RNNs display increased variance and reduced robustness as the number of hidden units increases.

remains well below recent benchmarks for the full AudioMNIST dataset, reporting accuracies up to 98% using more advanced models and architectures.^{23,31,32} Conventional RNN approaches typically rely on extracting features such as Mel spectrograms, log-Mel spectrograms, or MFCCs prior to training,³³ which reduces input dimensionality and helps mitigate classic RNN challenges like vanishing gradients and limited memory capacity. However, such preprocessing contradicts the design goal of acoustic neural networks, which aim to process raw sound intensities directly.

A natural extension of the RNN is to incorporate mechanisms that improve temporal memory, such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) architectures.^{21,22} However, these models rely on nonlinear activation functions such as sigmoid and tanh, which require finely controlled nonlinearities that are challenging to reproduce in acoustic media. Consequently, these architectures are unsuitable for direct physical realization. To overcome the limitations of standard RNNs while maintaining acoustic feasibility, we next investigate HSRNNs,²⁶ which provide improved robustness for processing long raw audio sequences.

B. HSRNNs

To extend the analysis beyond simple recurrent models, HSRNNs²⁶ were evaluated on the AudioMNIST dataset at a sampling rate of 1 kHz, comparing unconstrained and constrained variants across multiple hidden-unit configurations. This architecture introduces temporal downsampling between recurrent layers, enabling efficient modeling of long raw audio sequences while remaining compatible with the acoustic constraints imposed earlier.

Table II summarizes the results for the binary AudioMNIST subset (digits 0 and 1). Configurations are denoted by sequences such as 1-1-1 or 8-16-32, indicating the number of hidden units in the first, second, and third recurrent layers, respectively. Across most configurations, test accuracy remains comparable between the two model types, with the smallest architecture (1-1-1) performing poorly and showing the highest variability across runs. For mid-sized and large configurations, both networks reach accuracies above 85%, while the constrained model

often maintains a narrower gap between train and test accuracy. This reduced discrepancy indicates an implicit regularization effect arising from the non-negativity and absence of bias terms, which limit the model's expressiveness and mitigate overfitting.³⁴ In contrast, the unconstrained networks occasionally reach near-perfect train accuracy, suggesting a tendency toward overfitting.

Extending the evaluation to the full ten-digit AudioMNIST dataset yields the results shown in Tab. III. The overall trends observed in the binary case persist. The smallest constrained configuration (4-8-16) performs significantly worse than the corresponding unconstrained model, with a gap of roughly 18 percentage points, likely due to reduced representational capacity or suboptimal initialization.^{25,29,35} However, for larger networks, the constrained model matches or slightly exceeds the unconstrained model's test accuracy. The best constrained configuration (16-32-64) achieves a mean test accuracy of $66.8\% \pm 1.6\%$, representing the highest performance observed under acoustic constraints. Overall, these findings show that hierarchical subsampling improves performance for both constrained and unconstrained networks by enabling efficient modeling of long raw audio sequences.

Training stability in the constrained networks proved highly sensitive to weight initialization. For the representative 8-16-32 configuration, the corresponding performance shown in Fig. 3 indicates that stable behavior was achieved only for uniform weight initializations $\mathcal{U}(0, c)$ with c between 0.01 and 0.06. Larger values for c led to saturation and convergence failure, while smaller values caused vanishing gradients and unstable training. Although the exact range depends on the architecture and activation function, this trend highlights the need for carefully tuned initialization scales when enforcing non-negativity, as inappropriate values can effectively prevent learning.

Training in the constrained HSRNNs led to a pronounced concentration of small weights in the weight matrix. For instance, in the 16-32-64 configuration, approximately 12% of the weights were clamped to zero after convergence, while the remaining values were strongly biased toward low magnitudes. This distribution indicates a tendency toward sparsity, which arises naturally from the non-negativity constraint. Such behavior is consistent with prior work showing that non-

Hidden units	Unconstrained		Constrained	
	Train accuracy (%)	Test accuracy (%)	Train accuracy (%)	Test accuracy (%)
1-1-1	61.03 \pm 9.32	60.76 \pm 9.03	61.68 \pm 9.79	60.39 \pm 8.50
2-4-8	90.28 \pm 2.08	89.45 \pm 1.64	88.04 \pm 0.97	89.12 \pm 1.27
4-4-4	84.47 \pm 6.43	84.97 \pm 6.06	71.32 \pm 1.11	68.99 \pm 0.70
8-8-8	91.18 \pm 1.96	89.71 \pm 0.70	83.44 \pm 5.97	82.20 \pm 8.91
4-8-16	93.61 \pm 0.78	90.29 \pm 0.46	91.42 \pm 0.56	91.27 \pm 0.45
8-16-32	96.91 \pm 0.58	90.48 \pm 0.72	92.07 \pm 1.01	91.30 \pm 0.79
16-32-64	99.40 \pm 0.34	89.85 \pm 0.63	90.96 \pm 0.50	89.62 \pm 1.73

TABLE II. Mean train and test accuracies (%) with standard deviations for HSRNNs on the binary AudioMNIST dataset at a sampling rate of 1 kHz. Results are reported for unconstrained and physically constrained models across increasing network sizes. Performance generally improves with model capacity.

Hidden units	Unconstrained		Constrained	
	Train accuracy (%)	Test accuracy (%)	Train accuracy (%)	Test accuracy (%)
4-8-16	66.19 \pm 0.22	64.62 \pm 0.42	47.33 \pm 5.53	47.17 \pm 6.25
4-8-16-32	72.96 \pm 1.25	64.45 \pm 0.57	65.85 \pm 0.88	63.60 \pm 1.26
8-16-32	74.11 \pm 0.43	66.14 \pm 0.85	64.58 \pm 0.69	62.54 \pm 1.88
16-32-64	85.61 \pm 0.27	64.45 \pm 0.57	71.69 \pm 0.59	66.79 \pm 1.57

TABLE III. Mean train and test accuracies (%) with standard deviations for HSRNNs evaluated on the full ten-digit AudioMNIST dataset at a sampling rate of 1 kHz. Results are reported for unconstrained and physically constrained models across increasing network sizes. While constrained models underperform for the smallest configuration, both architectures achieve comparable performance at larger scales, with the constrained model reaching a maximum mean test accuracy of $66.8\% \pm 1.6\%$.

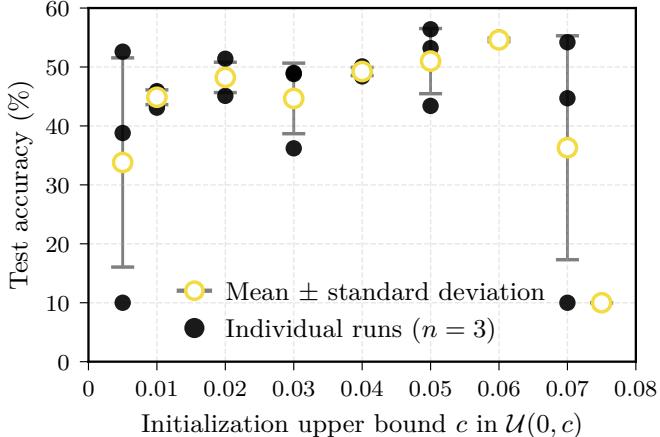


FIG. 3. Test accuracy (%) of the constrained HSRNN as a function of the upper bound c of the uniform weight initialization distribution $\mathcal{U}(0, c)$. Results are shown for the 8-16-32 architecture over three independent runs. Stable training is observed only for intermediate initialization scales ($0.01 \leq c \leq 0.06$), illustrating the high sensitivity of constrained models to weight initialization.

negative constraints promote compact or sparse-like representations.^{36,37} Such sparse structures could be advantageous for physical implementations, as they reduce the number of active transmission pathways.

In summary, the HSRNN achieves significantly higher accuracy than the simple RNN baseline while preserving compatibility with physical constraints. The constrained

variant benefits from an implicit regularization effect that improves generalization, although it remains sensitive to initialization and exhibits increased variability. These results establish the HSRNN as an effective architecture for processing long raw audio sequences under acoustic feasibility, laying the foundation for the subsequent integration of learnable frequency-selective filters in the SincHSRNN.

C. SincHSRNNs

To further enhance the classification of raw acoustic signals, we combined the hierarchical subsampling architecture with a learnable sinc-based filter front end,²⁷ yielding the SincHSRNN. The sinc filters act as parameterized bandpass filters, providing frequency-selective preprocessing that is both interpretable and physically realizable in acoustic systems. This hybrid design allows the model to learn frequency-domain representations directly from raw waveforms while capturing temporal dependencies at multiple time scales through the hierarchical recurrent structure.

Tables IV and V, together with Fig. 4 summarize the results across different hidden-unit configurations and sampling rates. The test accuracy increases consistently with higher sampling rates for both unconstrained and constrained variants, indicating that finer temporal resolution benefits the learnable filterbank. The confusion matrices in Fig. 4 further illustrate that the best-performing models achieve uniformly high per-class accu-

racy, with only minor misclassifications. The highest accuracies are achieved at 8 kHz with the 8-16-32-64 architecture, reaching 96.5% for the unconstrained and 95.1% for the constrained model.

For the unconstrained networks, the test accuracy varies little with hidden size (differences are below 3% across architectures at 8 kHz), suggesting that the performance is primarily limited by input resolution rather than model capacity. The constrained models, in contrast, exhibit stronger dependence on the model size: at 8 kHz, the accuracy rises from 85.3% to 94.7% when increasing from 8-16-32 to 16-32-64 hidden units. The largest performance gaps between constrained and unconstrained models occur at low sampling rates. At higher sampling rates and larger configurations, however, both models converge to nearly identical accuracies, differing by only a few percentage points.

Overall, the SincHSRNN substantially outperforms the pure HSRNN, achieving up to 95% test accuracy in the constrained setting compared to 67% for the best HSRNN model. The consistent improvement with the sampling rate indicates that the sinc filters leverage finer temporal resolution to extract frequency-selective features that are inaccessible to the raw recurrent model. The small remaining gap between constrained and unconstrained variants primarily reflects the reduced expressiveness imposed by non-negativity and the absence of bias terms, as discussed previously.^{25,35}

Notably, when evaluated on the full AudioMNIST dataset, the constrained SincHSRNN achieves performance approaching state-of-the-art results from convolutional neural network (CNN)-based architectures,^{23,31,32} which report accuracies approaching 98%. As only a limited set of hidden-layer configurations was explored in this study, further optimization of model capacity may lead to additional performance improvements. More recently, transformer-based models have also gained attention in raw audio processing due to their ability to model long-range dependencies.³⁸ However, transformer architectures typically involve millions of parameters,³⁸ which poses significant challenges for deployment in real-world acoustic systems. In contrast, the SincHSRNN presented in this thesis introduces a novel hierarchical RNN-based approach to raw audio classification that remains compact, using only about 13% of the parameters of comparable CNNs.³² This efficiency and simplicity make it a promising candidate for analog acoustic implementations.

Beyond its accuracy, the architecture offers intrinsic physical interpretability. The learnable sinc filters act as tunable bandpass elements, a standard function in acoustics, allowing a direct mapping between learned filter parameters and physical resonators or waveguide elements. In this sense, the SincHSRNN provides both a high-performing digital model and a plausible design blueprint for real-world acoustic neural networks.

D. Design principles and discussion

The results presented above reveal several general principles for the design of physically realizable acoustic neural networks. First, the restriction to non-negative weights and activations does not fundamentally limit the ability of the network to learn complex temporal patterns, provided that initialization and scaling are carefully controlled. The reduced overfitting in the constrained HSRNNs and SincHSRNNs demonstrates that non-negativity can act as a form of implicit regularization, promoting sparse representations. This sparsity is advantageous for physical implementations, as it reduces the number of active transmission pathways.

Second, hierarchical subsampling proved essential for effective learning on long raw audio sequences. By aggregating consecutive time steps, the HSRNN progressively reduces the sequence length between recurrent layers. This hierarchical compression alleviates vanishing and exploding gradient effects and lowers the computational load of the recurrent units. In an acoustic realization, the same operation could be achieved by splitting the signal into multiple propagation paths of different lengths and attenuation levels that recombine at a junction. Each path acts as a controllable delay line with a transmission coefficient corresponding to a learnable subsampling weight.

Third, the integration of learnable sinc filters at the input stage highlights the importance of preprocessing. The sinc-based front end allows the network to learn frequency-selective representations that correspond to acoustic bandpass elements. This not only enhances classification accuracy but also provides a physically meaningful decomposition of the input signal. In an experimental setting, such filters could be realized through arrays of resonators or waveguides tuned to specific frequency bands.

Together, these principles establish a practical framework for developing acoustic neural networks that combine physical feasibility with competitive performance through constrained connectivity, hierarchical temporal integration, and interpretable frequency-selective filtering.

IV. CONCLUSIONS

This study establishes a framework for designing and simulating acoustic neural networks, architectures that process information via the propagation of sound waves. By applying physically motivated constraints, including non-negative signals and weights, the omission of bias terms, and nonlinearities compatible with intensity-based, non-negative acoustic signals, we developed digital-twin models that serve as interpretable blueprints for potential analog implementations. Through a systematic progression from recurrent neural networks (RNNs) to hierarchical subsampling RNNs (HSRNNs) and the

Hidden units	1 kHz	2 kHz	8 kHz
8–16–32	74.15 ± 0.43	83.45 ± 1.26	92.21 ± 1.33
16–32–64	73.47 ± 0.36	83.72 ± 0.65	95.23 ± 1.81
8–16–32–64	73.94 ± 0.45	84.46 ± 0.53	96.51 ± 1.58

TABLE IV. Mean test accuracies (%) with standard deviations for unconstrained sinc-based hierarchical subsampling recurrent neural networks (SincHSRNNs) on the AudioMNIST dataset at sampling rates of 1, 2, and 8 kHz. Accuracy increases monotonically with sampling rate, indicating that higher temporal resolution enhances the effectiveness of the learned filterbank representation.

Hidden units	1 kHz	2 kHz	8 kHz
8–16–32	60.35 ± 2.55	74.74 ± 1.90	85.33 ± 1.68
16–32–64	69.61 ± 0.36	82.16 ± 1.42	94.70 ± 1.39
8–16–32–64	66.86 ± 0.82	81.78 ± 0.46	95.11 ± 1.67

TABLE V. Mean test accuracies (%) with standard deviations for physically constrained SincHSRNNs on the AudioMNIST dataset at sampling rates of 1, 2, and 8 kHz. Constrained models exhibit stronger dependence on network size, but approach the performance of unconstrained networks at higher sampling rates, achieving comparable accuracy for the largest configurations.

final SincHSRNN, we demonstrated that networks constrained by physical realizability can achieve high classification accuracy while preserving direct physical interpretability.

The results identify several guiding principles for the development of physically realizable acoustic neural networks. First, non-negativity constraints do not inherently limit representational capacity when initialization and scaling are appropriately controlled. Instead, they promote sparsity and act as an implicit regularization mechanism that enhances generalization. Second, hierarchical subsampling enables robust temporal processing of long sequences and could be physically realized through networks of acoustic delay lines and weighted signal paths. Third, the incorporation of learnable sinc filters introduces an interpretable, frequency-selective pre-processing stage that directly corresponds to acoustic bandpass elements. Together, these features allowed the SincHSRNN to reach up to 95% accuracy on the AudioMNIST dataset, approaching state-of-the-art digital performance while remaining compatible with passive acoustic components.

Beyond their computational performance, these findings highlight the feasibility of acoustic-based analog computation. The learned parameters of the models correspond to measurable material and geometric properties such as attenuation and transmission, establishing a direct link between machine learning representations and physical quantities. This mapping provides a systematic framework for developing low-power, wave-based neural processors that operate through physical signal propagation. In turn, it also enables applications in direct analog speech recognition and other on-device acoustic processing tasks that benefit from low-power, real-time operation.

Future research should focus on translating these digital-twin models into experimental realizations, for in-

stance by constructing small-scale acoustic networks using waveguide structures or resonator arrays that mimic the learned transmission coefficients and nonlinear attenuation effects. While the present work considers a purely passive acoustic system without external energy input, future implementations could in principle include active elements or nonlinear mechanisms capable of signal amplification, allowing effective weights greater than one. However, preliminary simulations indicated that such effects do not provide a meaningful benefit for training stability or performance and were therefore not pursued further here. Extending the simulations to include effects such as scattering, reflection, and noise would refine the mapping between digital parameters and real acoustic responses. On the algorithmic side, developing optimization schemes and initialization strategies specifically tailored for non-negative, physically constrained networks could further enhance stability and convergence. Experimentally, prototype implementations using acoustic metamaterials or other passive wave-based structures could enable direct analog inference on real sound signals, paving the way for low-power edge computing and passive acoustic sensing in environments where electronic systems face limitations.

Overall, this work provides both theoretical foundations and practical design principles for acoustic neural networks, demonstrating that neural computation can, in principle, be achieved through the physics of sound. These results place acoustic systems alongside optical and mechanical analogs as promising candidates for future energy-efficient, wave-guide-based neural computing.

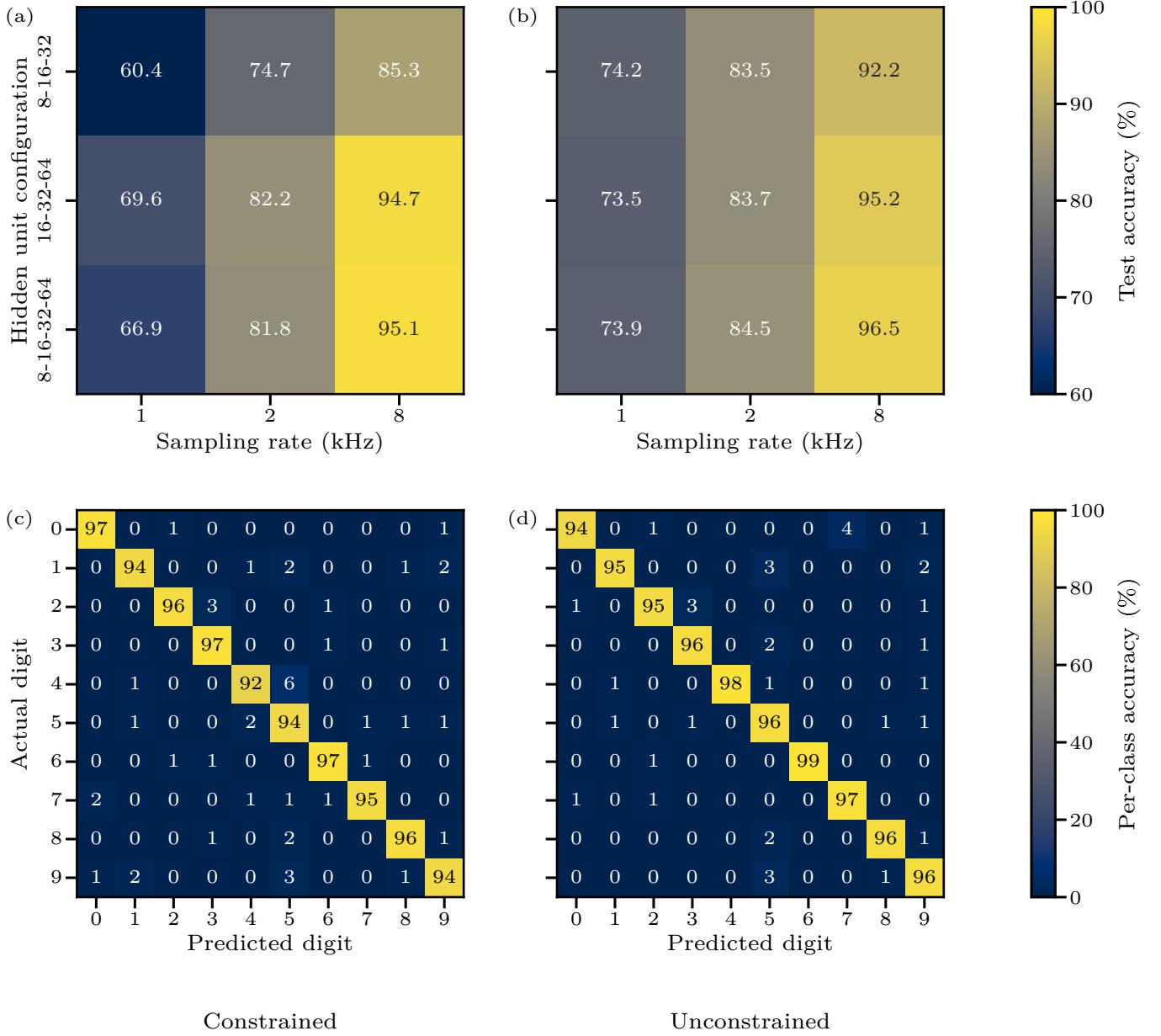


FIG. 4. (a), (b) Test accuracy heatmaps and (c), (d) confusion matrices for (a), (c) constrained and (b), (d) unconstrained SinchSRNNs on the full ten-digit AudioMNIST dataset. (a), (b) Test accuracy heatmaps show the mean test accuracies (%) across hidden-unit configurations and sampling rates for the constrained and unconstrained networks, respectively. Overall performance increases with sampling rate, and the gap between constrained and unconstrained models narrows at higher capacities. (c), (d) Confusion matrices for the best-performing architecture (8-16-32-64) evaluated at a sampling rate of 8 kHz, illustrating that both models achieve high per-class accuracy with only minor deviations.

SUPPLEMENTARY MATERIAL

See Supplementary Material at Ref. 39 for the complete source code used to implement and train all neural network models, as well as the raw data and plotting scripts corresponding to the figures presented in this work.

DATA AVAILABILITY

The data that support the findings of this study are available within the article and its Supplementary Material.

CONFLICTS OF INTEREST

dation) – 535275785 (R.W.); SFB 1459/2 2025 – 433682494 (R.W., M.R.).

There are no conflicts of interest to declare.

ACKNOWLEDGMENTS

R.W. and M.R. are funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foun-

Appendix A:

This appendix provides a comprehensive overview of the training parameters used for all neural network architectures evaluated in this work. For each network type, RNN, HSRNN, and SincHSRNN, both the unconstrained and constrained variants were trained using the parameters listed below. In constrained models, all weights were clamped to the range $[0, 1]$ after each optimization step to emulate acoustic attenuation. The specific hyperparameter values for each architecture are summarized in Tab. VI–VIII.

Parameter	Unconstrained network	Constrained network
Epochs	30	30
Batch size	64	64
Learning rate	0.001	0.001
Loss function	Cross-entropy loss	Cross-entropy loss
Optimizer	Adam	Adam
Bias	True	False
Weight initialization	$\mathcal{U}_{\text{Xavier}}(-a, a)$	$\mathcal{U}(0, c)$, $c \in [0.01, 0.1]$
Activation function	tanh	$ x - c $ or $\max(x - c, 0)$, $c \in [0.01, 0.1]$

TABLE VI. Training parameters for the RNNs used in the AudioMNIST binary classification task. The unconstrained network follows a standard Xavier uniform initialization $\mathcal{U}_{\text{Xavier}}(-a, a)$ and employs a tanh activation. The constrained network uses non-negative weights sampled from $\mathcal{U}(0, c)$ with $c \in [0.01, 0.1]$ and replaces the activation with offset absolute or offset ReLU functions $|x - c|$ or $\max(x - c, 0)$, respectively.

Parameter	Unconstrained network	Constrained network
Epochs	50	50-200
Batch size	64	64
Learning rate	0.001	0.001
Hidden layers	3	3
Loss function	Cross-entropy loss	Cross-entropy loss
Optimizer	Adam	Adam
Bias	True	False
Weight initialization	$\mathcal{U}_{\text{Xavier}}(-a, a)$	$\mathcal{U}(0, c)$, $c \in [0.05, 0.3]$
Activation function	tanh	$ x - c $ or $\max(x - c, 0)$, $c \in [0.05, 0.3]$

TABLE VII. Training parameters for the HSRNNs used in the AudioMNIST classification task. The unconstrained network employs standard Xavier uniform initialization $\mathcal{U}_{\text{Xavier}}(-a, a)$ with tanh activations. The constrained variant enforces non-negative weights drawn from $\mathcal{U}(0, c)$ with $c \in [0.05, 0.3]$, replaces the activation by offset absolute or offset ReLU functions $|x - c|$ or $\max(x - c, 0)$, and removes bias terms. Longer training schedules (50-200 epochs) were required for convergence under these constraints.

Parameter	Unconstrained network	Constrained network
Epochs	40	60-95
Batch size	64	64
Learning rate	0.001, 0.0001	0.001, 0.0001
Hidden layers	3-4	3-4
Loss function	Cross-entropy loss	Cross-entropy loss
Optimizer	Adam	Adam
Bias	True	False
Subsampling factor	8	8
Sinc kernel size	101	101
Sinc output channels	5	5
Weight initialization	$\mathcal{U}_{\text{Xavier}}(-a, a)$	$ \mathcal{U}_{\text{Xavier}}(-a, a) $, gain = 0.13
Activation function	tanh	$ x - c $, $c \in [0.9, 1]$

TABLE VIII. Training parameters for the SincHSRNN used in the AudioMNIST classification task. The unconstrained network follows standard Xavier uniform initialization $\mathcal{U}_{\text{Xavier}}(-a, a)$ with tanh activations. The constrained variant employs non-negative weights obtained from the absolute Xavier uniform distribution $|\mathcal{U}_{\text{Xavier}}(-a, a)|$ with a gain factor of 0.13, and uses offset absolute activations $|x - c|$ with $c \in [0.9, 1]$. Bias terms were omitted. Training was performed for 60-95 epochs, with the final 10 epochs at a reduced learning rate 0.0001 for fine-tuning.

- ¹ Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature* **521**, 436–444 (2015).
- ² N. Thompson, K. Greenewald, K. Lee, and G. F. Manso, “The Computational Limits of Deep Learning,” in *Ninth Computing within Limits* (2023).
- ³ T. N. Theis and H.-S. P. Wong, “The end of Moore’s law: A new beginning for information technology,” *Computing in Science & Engineering* **19**, 41–50 (2017).
- ⁴ D. Marković, A. Mizrahi, D. Querlioz, and J. Grollier, “Physics for neuromorphic computing,” *Nature Reviews Physics* **2**, 499–510 (2020).
- ⁵ J. Weng, Y. Ding, C. Hu, X.-F. Zhu, B. Liang, J. Yang, and J. Cheng, “Meta-neural-network for real-time and passive deep-learning-based object recognition,” *Nature Communications* **11**, 6309 (2020).
- ⁶ T. Jiang, “Metamaterial-based analog recurrent neural network toward machine intelligence,” *Physical Review Applied* **19**, 064065 (2023).
- ⁷ R. H. Lee, E. A. B. Mulder, and J. B. Hopkins, “Mechanical neural networks: Architected materials that learn behaviors,” *Science Robotics* **7**, eabdq7278 (2022).
- ⁸ A. Silva, F. Monticone, G. Castaldi, V. Galdi, A. Alù, and N. Engheta, “Performing mathematical operations with metamaterials,” *Science* **343**, 160–163 (2014).
- ⁹ S. Zuo, Q. Wei, Y. Tian, Y. Cheng, and X. Liu, “Acoustic analog computing system based on labyrinthine metasurfaces,” *Scientific Reports* **8**, 10103 (2018).
- ¹⁰ F. Zangeneh-Nejad, D. L. Sounas, A. Alù, and R. Fleury, “Analogue computing with metamaterials,” *Nature Reviews Materials* **6**, 207–225 (2021).
- ¹¹ X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, “All-optical machine learning using diffractive deep neural networks,” *Science* **361**, 1004–1008 (2018).
- ¹² R. Xu, P. Lv, F. Xu, and Y. Shi, “A survey of approaches for implementing optical neural networks,” *Optics & Laser Technology* **136**, 106787 (2021).
- ¹³ T. W. Hughes, I. A. D. Williamson, M. Minkov, and S. Fan, “Wave physics as an analog recurrent neural network,” *Science Advances* **5**, eaay6946 (2019).
- ¹⁴ M. Zolfagharinejad, J. Büchel, L. Cassola, S. Kinge, G. S. Syed, A. Sebastian, and W. G. van der Wiel, “Analogue speech recognition based on physical computing,” *Nature* **645**, 886–892 (2025).
- ¹⁵ X. Fang, W. Lacarbonara, and L. Cheng, “Advances in nonlinear acoustic/elastic metamaterials and metastructures,” *Nonlinear Dynamics* **113**, 23787–23814 (2025).
- ¹⁶ S. Fiore, G. Finocchio, R. Zivieri, M. Chiappini, and F. Garesci, “Wave amplitude decay driven by anharmonic potential in nonlinear mass-in-mass systems,” *Applied Physics Letters* **117**, 124101 (2020).
- ¹⁷ N. Mork, M. D. Fronk, M. B. Sinclair, and M. J. Leamy, “Nonlinear hierarchical unit cell for passive, amplitude-dependent filtering of acoustic waves,” *Extreme Mechanics Letters* **57**, 101915 (2022).
- ¹⁸ Y. Huang, X. Wang, X. Gong, H. Wu, D. Zhang, and D. Zhang, “Contact nonlinear acoustic diode,” *Scientific Reports* **10**, 2564 (2020).
- ¹⁹ H. Xiao, C. Zhang, Y. Li, D. Ta, and X. Jiang, “Acoustic logic networks realized with non-Hermitian and non-local metagratings,” *Physical Review Applied* **22**, 024048 (2024).
- ²⁰ S. Li and X. Mao, “Training all-mechanical neural networks for task learning through in situ backpropagation,” *Nature Communications* **15**, 10528 (2024).
- ²¹ S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation* **9**, 1735–1780 (1997).
- ²² K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*

- ing (EMNLP)* (Association for Computational Linguistics, 2014) pp. 1724–1734.
- ²³ S. Becker, J. Vielhaben, M. Ackermann, K.-R. Müller, S. Lapuschkin, and W. Samek, “AudioMNIST: Exploring explainable artificial intelligence for audio analysis on a simple benchmark,” *Journal of the Franklin Institute* **361**, 418–428 (2024).
- ²⁴ T. Wang, S.-Y. Ma, L. G. Wright, T. Onodera, B. C. Richard, and P. L. McMahon, “An optical neural network using less than 1 photon per multiplication,” *Nature Communications* **13**, 123 (2022).
- ²⁵ M. Becker, D. Drees, F. Brückerhoff-Plückelmann, C. Schuck, W. Pernice, and B. Risse, “Activation functions in non-negative neural networks,” in *Machine Learning and the Physical Sciences Workshop, (NeurIPS)* (2023).
- ²⁶ A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Studies in Computational Intelligence, Vol. 385 (Springer, Berlin, Heidelberg, 2012).
- ²⁷ M. Ravanelli and Y. Bengio, “Speaker recognition from raw waveform with sincnet,” in *2018 IEEE Spoken Language Technology Workshop (SLT)* (2018) pp. 1021–1028.
- ²⁸ D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations ICLR* (2015).
- ²⁹ P.-J. Hoedt and G. Klambauer, “Principled weight initialisation for input-convex neural networks,” in *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)* (2023).
- ³⁰ B. Amos, L. Xu, and J. Z. Kolter, “Input convex neural networks,” in *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 70 (PMLR, 2017) pp. 146–155.
- ³¹ A. M. Tripathi and K. Paul, “When sub-band features meet attention mechanism while knowledge distillation for sound classification,” *Applied Acoustics* **195**, 108813 (2022).
- ³² H. Peic Tukuljac, B. Ricaud, N. Aspert, and L. Colbois, “Learnable filter-banks for CNN-based audio applications,” in *Proceedings of the Northern Lights Deep Learning Workshop; 3* (2022).
- ³³ K. Zaman, M. Sah, C. Direkoglu, and M. Unoki, “A survey of audio classification using deep learning,” *IEEE Access* **11**, 106620–106649 (2023).
- ³⁴ S. Sivaprasad, A. Singh, N. Manwani, and V. Gandhi, “The curious case of convex neural networks,” in *Machine Learning and Knowledge Discovery in Databases. Research Track* (Springer International Publishing, Cham, 2021) pp. 738–754.
- ³⁵ M. Kirtas, N. Passalis, N. Pleros, and A. Tefas, “Non-negative isomorphic neural networks for photonic neuromorphic accelerators,” arXiv preprint arXiv:2310.01084 (2023).
- ³⁶ B. O. Ayinde and J. M. Zurada, “Deep learning of constrained autoencoders for enhanced understanding of data,” *IEEE Transactions on Neural Networks and Learning Systems* **29**, 3969–3979 (2018).
- ³⁷ J. Chorowski and J. M. Zurada, “Learning understandable neural networks with nonnegative weight constraints,” *IEEE Transactions on Neural Networks and Learning Systems* **26**, 62–69 (2015).
- ³⁸ S. Latif, S. A. M. Zaidi, H. Cuayáhuitl, F. Shamshad, M. Shoukat, M. Usama, and J. Qadir, “Transformers in speech processing: Overcoming challenges and paving the future,” *Computer Science Review* **58**, 100768 (2025).
- ³⁹ I. Kalthoff, M. Rey, and R. Wittkowski, “Supplementary code and data,” Zenodo, <https://doi.org/10.5281/zenodo.17587626> (2025).