Wave Physics as an Analog Recurrent Neural Network

Tyler W. Hughes,[1,*] Ian A. D. Williamson,[2,*] Momchil Minkov,[2] and Shanhui Fan[2,†]

[1] *Department of Applied Physics, Stanford University, Stanford, CA 94305, USA*
[2] *Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA*

Analog machine learning hardware platforms promise to be faster and more energy-efficient than their digital counterparts. Wave physics, as found in acoustics and optics, is a natural candidate for building analog processors for time-varying signals. Here we identify a mapping between the dynamics of wave physics, and the computation in recurrent neural networks. This mapping indicates that physical wave systems can be trained to learn complex features in temporal data, using standard training techniques for neural networks. As a demonstration, we show that an inversely-designed inhomogeneous medium can perform vowel classification on raw audio data by simple propagation of waves through such a medium, achieving performance that is comparable to a standard digital implementation of a recurrent neural network. These findings pave the way for a new class of analog machine learning platforms, capable of fast and efficient processing of information in its native domain.

Recently, machine learning has had significant success in performing complex information processing tasks, such as computer vision [1] and machine translation [2], which were previously intractable through traditional methods. However, the computing requirements of these applications is increasing exponentially, motivating efforts to develop new, specialized hardware platforms for fast and efficient execution of machine learning models. Among these are *neuromorphic* hardware platforms [3–5], with architectures that mimic the biological circuitry of the brain. Furthermore, analog computing platforms, which use the natural evolution of a continuous physical system to perform calculations, are also emerging as important direction for implementation of machine learning [6–9].

Here, we identify a mapping between the dynamics of wave-based physical phenomena, such as acoustics and optics, and the computation in a recurrent neural network (RNN). RNNs are one of the most important machine learning models and have been widely used to perform tasks such as natural language processing [10] and time-series prediction [11–13]. Here we show that wave-based physical systems can be trained to operate as an RNN, and as a result can *passively* process signals and information in their native domain, without analog-to-digital conversion, which should result in a significant gain in speed and a reduction in power consumption.

An RNN converts a sequence of inputs into a sequence of outputs by applying the same basic operation to each member of the input sequence in a step-by-step fashion (Fig 1A). Memory of previous time steps is encoded into the RNN's *hidden state*, which is updated at each step. The information in the hidden state allows for the RNN to learn temporal structure and long-range dependencies in data [14, 15]. At a given time step, t, the RNN operates on the current input vector in the sequence, $\boldsymbol{x}_t$, and the hidden state vector from the previous step, $\boldsymbol{h}_{t-1}$, to produce an output vector, $\boldsymbol{y}_t$, as well as an updated hidden state, $\boldsymbol{h}_t$. While many variations of RNNs exist, a common implementation [16] is described by the following update equations

$$\boldsymbol{h}_t = \sigma^{(h)}\Big(\boldsymbol{W}^{(h)} \cdot \boldsymbol{h}_{t-1} + \boldsymbol{W}^{(x)} \cdot \boldsymbol{x}_t\Big) \tag{1}$$

$$\boldsymbol{y}_t = \sigma^{(y)}\Big(\boldsymbol{W}^{(y)} \cdot \boldsymbol{h}_t\Big), \tag{2}$$

which are diagrammed in Fig. 1B. The dense matrices defined by $\boldsymbol{W}^{(h)}$, $\boldsymbol{W}^{(x)}$, and $\boldsymbol{W}^{(y)}$ are optimized during training while $\sigma^{(h)}(\cdot)$ and $\sigma^{(y)}(\cdot)$ are nonlinear activation functions. The operation defined by Eq. 1 and Eq. 2, when applied to each element of an input sequence, can be described by the directed graph shown in Fig. 1C.

We now discuss the connection between the dynamics in the RNN as described by Eqs. 1 and 2, and the dynamics of wave-based physical systems. As an illustration, the dynamics of a scalar wave field distribution $u(x, y, z)$ are governed by the second-order partial differential equation (Fig. 1D),

$$\frac{\partial^2 u}{\partial t^2} - c^2 \cdot \nabla^2 u = f, \tag{3}$$

where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ is the Laplacian operator, $c = c(x, y, z)$ is the spatial distribution of the local wave speed. For sufficiently large fields, c may additionally depend nonlinearly on the local wave amplitude. $f = f(x, y, z, t)$ is a source term. A finite-difference discretization of Eq. 3, with a temporal step size of Δt, results in the recurrence relation,

$$\frac{u_{t+1} - 2u_t + u_{t-1}}{\Delta t^2} - c^2 \cdot \nabla^2 u_t = f_t. \tag{4}$$

Here, the subscript in $(\cdot)_t$ is used to indicate the value of a scalar field at time step t. The wave system's *hidden state* is defined as the concatenation of the field distributions at the current and immediately preceding time steps, $\boldsymbol{h}_t \equiv [\boldsymbol{u}_t,\ \boldsymbol{u}_{t-1}]^T$, where $\boldsymbol{u}_t$ and $\boldsymbol{u}_{t-1}$ are vectors given by the *flattened* fields, u_t and u_{t-1}, discretized in the spatial

* These authors contributed equally to this work.
† shanhui@stanford.edu

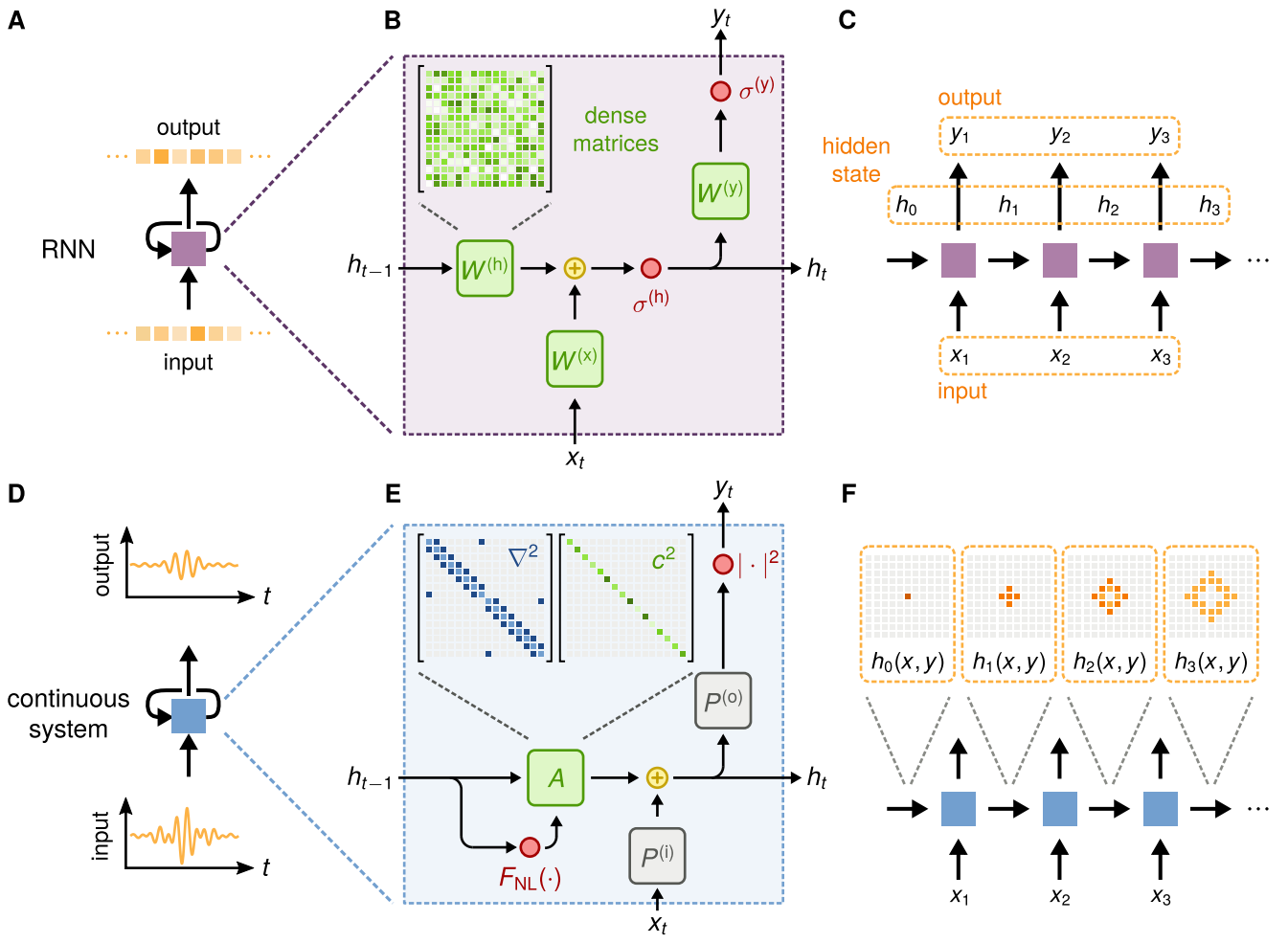

Figure 1. **Conceptual comparison of a standard recurrent neural network and a wave-based physcal system.** (**A**) Diagram of a recurrent neural network (RNN) cell operating on a discrete input sequence and producing a discrete output sequence. (**B**) Internal components of the RNN cell, consisting of trainable dense matrices $\boldsymbol{W}^{(h)}$, $\boldsymbol{W}^{(x)}$, and $\boldsymbol{W}^{(y)}$. Activation functions for the hidden state and output are represented by $\sigma^{(h)}$ and $\sigma^{(y)}$, respectively. (**C**) Diagram of the directed graph of the RNN cell. (**D**) Diagram of a recurrent representation of a continuous physical system operating on a continuous input sequence and producing a continuous output sequence. (**E**) Internal components of the recurrence relation for the wave equation when discretized using finite differences. (**F**) Diagram of the directed graph of discrete time steps of the continuous physical system.

domain. Then, the update of the wave equation may be written as

$$\boldsymbol{h}_t = \boldsymbol{A}(\boldsymbol{h}_{t-1}) \cdot \boldsymbol{h}_{t-1} + \boldsymbol{P}^{(\mathrm{i})} \cdot \boldsymbol{x}_t \tag{5}$$

$$\boldsymbol{y}_t = \left| \boldsymbol{P}^{(\mathrm{o})} \cdot \boldsymbol{h}_t \right|^2, \tag{6}$$

where the sparse matrix, $\boldsymbol{A}$, describes the update of the wave fields $\boldsymbol{u}_t$ and $\boldsymbol{u}_{t-1}$ without a source (Fig. 1E). The derivation of Eq. 5 and 6 are given in supplementary information section S1.

The dependence of $\boldsymbol{A}$ on $\boldsymbol{h}_{t-1}$ can be achieved through an intensity-dependent wave speed of the form $c(x, y) = c_{\mathrm{linear}} + c_{\mathrm{nonlinear}} \cdot |u_t(x, y)|^2$, where $c_{\mathrm{nonlinear}}$ is exhibited in regions containing nonlinear materials. In practice, this class of nonlinearity is encountered in a variety of wave physics, such as shallow water waves [17] or nonlinear optics via the Kerr effect [18]. Similarly to the $\sigma^{(y)}(\cdot)$ activation function in a standard RNN, a nonlinear relationship between the hidden state, $\boldsymbol{h}_t$, and the output, $\boldsymbol{y}_t$, of the wave equation is typical in wave physics since the output usually corresponds to a measurement of the wave intensity, as we assume here for Eq. 6.

Like the standard RNN, the connections between the hidden state and the input and output of the wave equation are also defined by linear operators, given by $\boldsymbol{P}^{(\mathrm{i})}$ and $\boldsymbol{P}^{(\mathrm{o})}$. These matrices define the injection and measuring points within the spatial domain. Unlike the standard RNN, where the input and output matrices are dense, the input and output matrices of the wave equation are usually sparse and, moreover, unchanged by the

training process. Additional information on the construction of these matrices is given in supplementary information section S2.

We take the wave speed distribution, $c(x, y, z)$ as trainable parameters, which are optimized for a given machine learning task, physically corresponding to a patterning of materials within the domain. Thus, when modeled numerically in discrete time (Fig. 1E), the wave equation defines an operation which maps into that of an RNN (Fig. 1B). Similarly to the RNN, the full time dynamics of the wave equation may be represented as a directed graph (Fig. 1F).

We now demonstrate how the dynamics of the wave equation can be trained to classify vowels through the construction of an inhomogeneous material distribution. For this task, we utilize a dataset consisting of 930 raw audio recordings of 10 vowel classes from 45 different male speakers and 48 different female speakers [19]. For our learning task, we select a subset of 279 recordings corresponding to three vowel classes, represented by the vowel sounds *ae*, *ei*, and *iy*, as contained in the words h*a*d, h*aye*d, and h*ee*d, respectively (Fig. 2A).

The physical layout of the vowel recognition system consists of a two-dimensional domain in the x-y plane, infinitely extended along the z-direction (Fig. 2B). The audio waveform of each vowel, represented by $\boldsymbol{x}^{(i)}$, is injected by a source at a single grid cell on the left side of the domain, emitting waveforms which propagate through a central region with a trainable distribution of the wave speed, indicated by the light green region in Fig. 2B. Three probe points are defined on the right hand side of this region, each assigned to one of the three vowel classes. To determine the system's output, $\boldsymbol{y}^{(i)}$, the time-integrated power at each probe is measured (Fig. 2C). After the simulation evolves for the full duration of the vowel recording, this integral gives a non-negative vector of length 3, which is then normalized by its sum and interpreted as the system's predicted probability distribution over the vowel classes. An absorbing boundary region, represented by the dark gray region in Fig. 2B, is included in the simulation to prevent energy from building up inside the computational domain. The derivation of the discretized wave equation with a dampening term is given in supplementary information section S1.

For the purposes of our numerical demonstration, we consider binarized systems consisting of two materials: a background material with a normalized wave speed $c_0 = 1.0$, and a second material with $c_1 = 0.5$. We assume that the second material has a nonlinear parameter, $c_{\text{nonlinear}} = -30$, while the background material has a linear response. In practice, the wave speeds would be modified to correspond to different materials being used. For example, in an acoustic setting the material distribution could consist of air, where the sound speed is 331 m/s, and porous silicone rubber, where the sound speed is 150 m/s [20]. The initial distribution of the wave speed consists of a uniform region of material with a speed which is midway between those of the two materials (Fig. 2D). This choice of starting structure allows for the optimizer to shift the density of each pixel towards either one of the two materials to produce a binarized structure consisting of only those two materials. To train the system, we perform back-propagation through the model of the wave equation to compute the gradient of the cross entropy loss function of the measured outputs with respect to the density of material in each pixel of the trainable region. Interestingly, this approach is mathematically equivalent to the *adjoint method* [21], which is widely used for inverse design [21–23]. Then, we use this gradient information update the material density using the the Adam optimization algorithm [24], repeating until convergence on a final structure (Fig. 2D).

The confusion matrices over the training and testing sets for the starting structure are shown in Fig. 3A and Fig. 3B, averaged over five cross-validated training runs. Here, the confusion matrix defines the percentage of correctly predicted vowels along its diagonal entries and the percentage of incorrectly predicted vowels for each class in its off-diagonal entries. Clearly the starting structure can not perform the recognition task. Fig. 3C and Fig. 3D show the final confusion matrices after optimization for the testing and training sets, averaged over five cross validated training runs. The trained confusion matrices are diagonally dominant, indicating that the structure can indeed perform vowel recognition. More information on the training procedure and numerical modeling is provided in supplementary information sections S3 and S4, respectively.

Fig. 3E and Fig. 3F show the cross entropy loss value and the prediction accuracy, respectively, as a function of the training epoch over the testing and training datasets, where the solid line indicates the mean and the shaded region corresponds to the standard deviation over the cross-validated training runs. Interestingly, we observe that the first epoch results in the largest reduction of the loss function and the largest gain in prediction accuracy. From Fig. 3F we see that the system obtains a mean accuracy of 92.6% $\pm$ 1.1% over the training dataset and a mean accuracy of 86.3% $\pm$ 4.3% over the testing dataset. From Fig. 3C and Fig. 3D we observe that the system attains near perfect prediction performance on the *ae* vowel and is able to differentiate the *iy* vowel from the *ei* vowel, but with less accuracy, especially in unseen samples from the testing dataset. Fig. 3G, Fig. 3H, and Fig. 3I show the distribution of the integrated field intensity, $\sum_t |u_t(x, y)|^2$, when a representative sample from each vowel class is injected into the trained structure. We thus provide visual confirmation that the optimization procedure produces a structure which routes the majority of the signal energy to the correct probe. As a performance benchmark, a conventional RNN was trained on the same task, achieving comparable classification accuracy to that of the wave equation. However, a larger number of free parameters was required. Additionally, we observed that a comparable classification accuracy was obtained when training a linear wave equation. More details on the per-

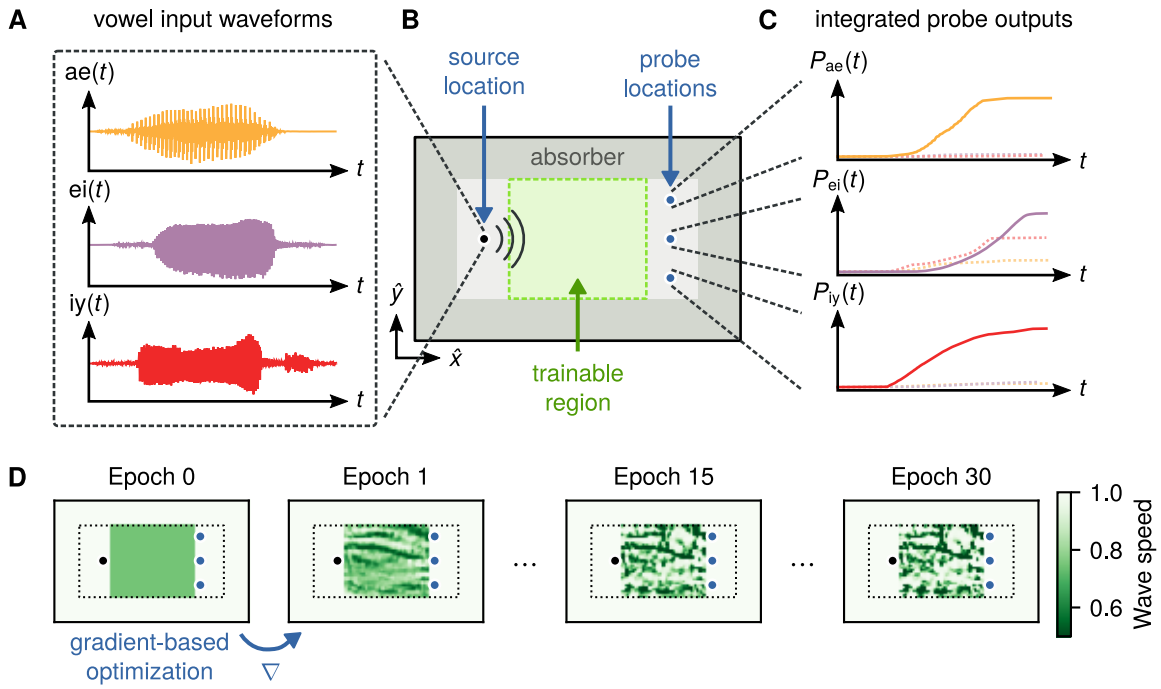

Figure 2. **Schematic of the vowel recognition system and the training procedure.** (**A**) Raw audio waveforms of spoken vowel samples from three classes. (**B**) Layout of the vowel recognition system. Vowel samples are independently injected at the source, located at the left of the domain, and propagate through the center region, indicated in green, where a material distribution is optimized during training. The dark gray region represents an absorbing boundary layer. (**C**) For classification, the time-integrated power at each probe is measured and normalized to be interpreted as a probability distribution over the vowel classes. (**D**) Using automatic differentiation, the gradient of the loss function with respect to the density of material in the green region is computed. The material density is updated iteratively, using gradient-based stochastic optimization techniques, until convergence.

formance are provided in supplementary information section S5.

The wave-based RNN presented here has a number of favorable qualities that make it a good candidate for processing sequential data. Unlike the standard RNN, the update of the wave equation from one time step to the next enforces a nearest-neighbor coupling between elements of the hidden state through the Laplacian operator, which is represented by a sparse matrix in Fig. 1E. This is a direct consequence of the fact that the wave equation is a hyperbolic partial differential equation in which information propagates with a finite velocity. Thus, the size of the analog RNN's hidden state, and therefore its memory capacity, is directly determined by the size of the propagation medium. Additionally, unlike the conventional RNN, the wave equation enforces an energy conservation constraint, preventing unbounded growth of the norm of the hidden state and the output signal. In contrast, the unconstrained dense matrices defining the update relationship of the standard RNN lead to vanishing and exploding gradients, which pose a major challenge for training traditional RNNs [25].

We have shown that the dynamics of the wave equation are conceptually equivalent to those of a recurrent neural network. This conceptual connection opens up the opportunity for a new class of analog hardware platform, in which evolving time dynamics play a significant role in both the physics and the dataset. While we have focused on a specific example of the scalar wave equation, our results apply broadly to other wave-like physics. Such an approach of using physics to perform computation [9, 26–30] may inspire a new platform for analog machine learning devices, with the potential to perform computation far more naturally and efficiently than their digital counterparts. The generality of the approach further suggests that many physical systems may be attractive candidates for performing RNN-like computation on naturally occurring sequences, such as optical, acoustic, or seismic signals.

ACKNOWLEDGEMENTS

Funding: This work was supported by a Vannevar Bush Faculty Fellowship (Grant № N00014-17-1-3030) from the U.S. Department of Defense, by the Gordon and Betty Moore Foundation (Grant № GBMF4744), by a MURI grant from the U.S. Air Force Office of Scientific Research (Grant № FA9550-17-1-0002), and by the Swiss National Science Foundation (Project №

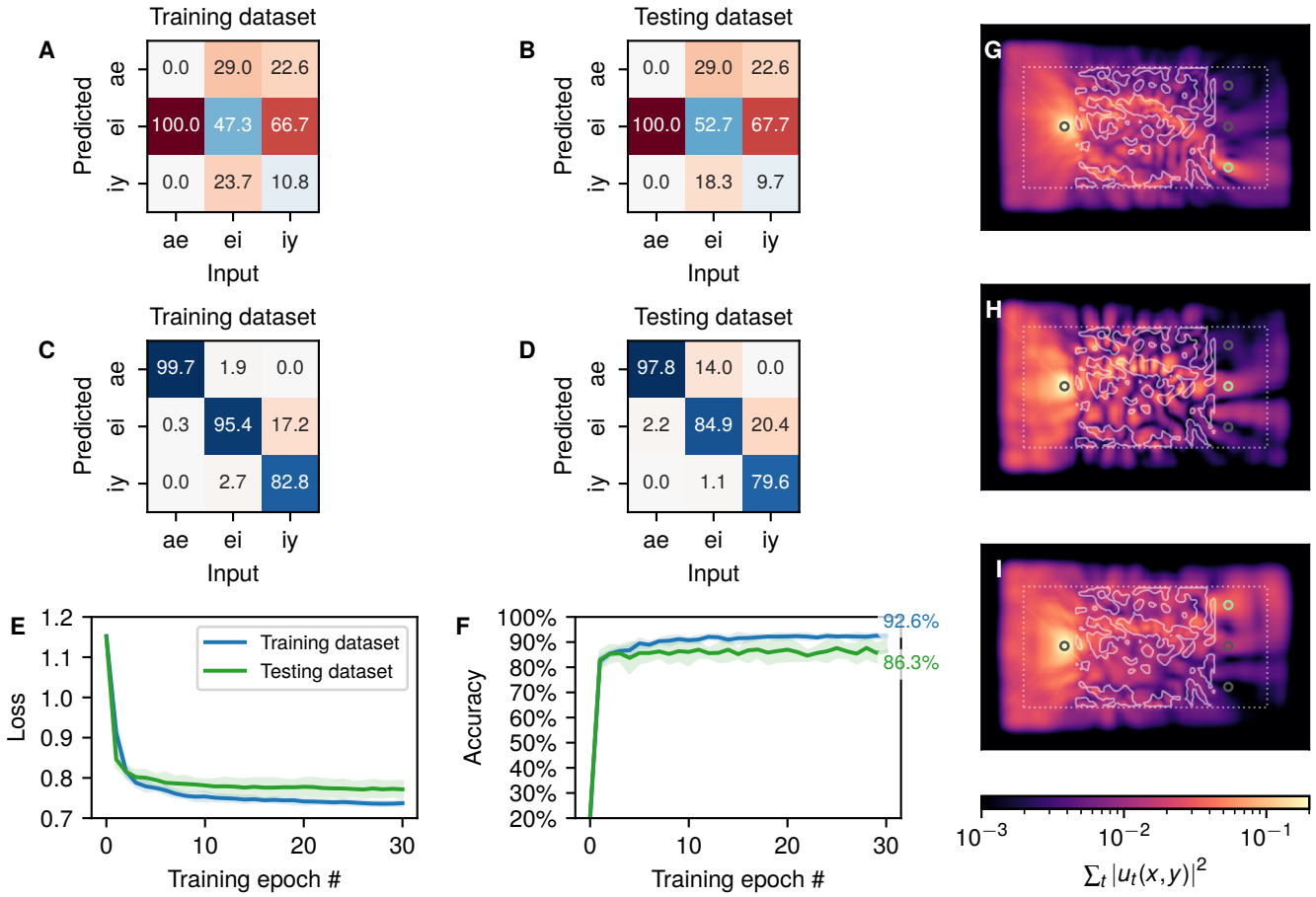

Figure 3. **Vowel recognition system training results.** Confusion matrix over the training and testing datasets for the initial structure (**A**),(**B**) and final structure (**C**),(**D**), indicating the percentage or correct (diagonal) and incorrect (off-diagonal). Cross validated training results showing the mean (sold line) and standard deviation (shaded region) of the (**E**) cross entropy loss and (**F**) prediction accuracy over 20 training epochs and 5 folds of the dataset, which consists of a total of 272 total vowel samples of male and female speakers. (**G**)-(**I**) The time-integrated intensity distribution for a randomly selected input (**G**) *ae* vowel, (**H**) *ei* vowel, and (**I**) *iy* vowel.

P300P2_177721). **Author contributions:** T.W.H conceived of the idea with input from I.A.D.W. and M.M.. The software for performing numerical simulations and training of the wave equation was written by I.A.D.W. with input from T.W.H. and M.M.. The model for the standard RNN was developed and trained by M.M.. S.F. supervised the project. All authors contributed to analyzing the results and writing the manuscript. **Competing interests:** The authors have jointly filed for a provisional patent on the idea. The authors declare no other competing interests. **Data and materials availability:** The code for performing numerical simulations and training of the wave equation, as well as generating the figures presented in this paper are available online at `http://www.github.com/fancompute/wavetorch/`.

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," International Journal of Computer Vision **115**, 211–252 (2015).

[2] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems 27*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Curran Associates, Inc., 2014) pp. 3104–3112.

[3] Jeffrey M Shainline, Sonia M Buckley, Richard P Mirin, and Sae Woo Nam, "Superconducting optoelectronic circuits for neuromorphic computing," Physical Review Applied **7**, 034013 (2017).

[4] Alexander N. Tait, Thomas Ferreira de Lima, Ellen Zhou, Allie X. Wu, Mitchell A. Nahmias, Bhavin J. Shastri, and Paul R. Prucnal, "Neuromorphic photonic networks

using silicon photonic weight banks," Scientific Reports **7**, 7430 (2017).

[5] Miguel Romera, Philippe Talatchian, Sumito Tsunegi, Flavio Abreu Araujo, Vincent Cros, Paolo Bortolotti, Juan Trastoy, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Maxence Ernoult, Damir Vodenicarevic, Tifenn Hirtzlin, Nicolas Locatelli, Damien Querlioz, and Julie Grollier, "Vowel recognition with four coupled spin-torque nano-oscillators," Nature **563**, 230 (2018).

[6] Yichen Shen, Nicholas C. Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, and Marin Soljačić, "Deep learning with coherent nanophotonic circuits," Nature Photonics **11**, 441–446 (2017).

[7] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd, "Quantum machine learning," Nature **549**, 195–202 (2017).

[8] Floris Laporte, Andrew Katumba, Joni Dambre, and Peter Bienstman, "Numerical demonstration of neuromorphic computing with photonic crystal cavities," Optics express **26**, 7955–7964 (2018).

[9] Xing Lin, Yair Rivenson, Nezih T Yardimci, Muhammed Veli, Yi Luo, Mona Jarrahi, and Aydogan Ozcan, "All-optical machine learning using diffractive deep neural networks," Science **361**, 1004–1008 (2018).

[10] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu, "Recurrent neural networks for language understanding." in *Interspeech* (2013) pp. 2524–2528.

[11] Michael Hsken and Peter Stagge, "Recurrent neural networks for time series classification," Neurocomputing **50**, 223–235 (2003).

[12] Georg Dorffner, "Neural Networks for Time Series Processing," Neural Network World **6**, 447–468 (1996).

[13] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," IEEE Transactions on Neural Networks **5**, 240–254 (1994).

[14] Jeffrey L Elman, "Finding structure in time," Cognitive Science **14**, 179–211 (1990).

[15] Michael I Jordan, "Serial order: A parallel distributed processing approach," in *Advances in psychology*, Vol. 121 (Elsevier, 1997) pp. 471–495.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning* (MIT Press, 2016).

[17] F. Ursell, "The long-wave paradox in the theory of gravity waves," Mathematical Proceedings of the Cambridge Philosophical Society **49**, 685 (1953).

[18] Robert W. Boyd, *Nonlinear Optics* (Academic Press, 2008).

[19] James Hillenbrand, Laura A. Getty, Michael J. Clark, and Kimberlee Wheeler, "Acoustic characteristics of American English vowels," The Journal of the Acoustical Society of America **97**, 3099–3111 (1995).

[20] Abdoulaye Ba, Artem Kovalenko, Christophe Aristégui, Olivier Mondain-Monval, and Thomas Brunet, "Soft porous silicone rubbers with ultra-low sound speeds in acoustic metamaterials," Scientific Reports **7**, 40106 (2017).

[21] Tyler W. Hughes, Momchil Minkov, Yu Shi, and Shanhui Fan, "Training of photonic neural networks through in situ backpropagation and gradient measurement," Optica **5**, 864–871 (2018).

[22] Sean Molesky, Zin Lin, Alexander Y Piggott, Weiliang Jin, Jelena Vucković, and Alejandro W Rodriguez, "Inverse design in nanophotonics," Nature Photonics **12**, 659 (2018).

[23] Y. Elesin, B. S. Lazarov, J. S. Jensen, and O. Sigmund, "Design of robust and efficient photonic switches using topology optimization," Photonics and Nanostructures - Fundamentals and Applications **10**, 153–165 (2012).

[24] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 (2014).

[25] Li Jing, Yichen Shen, Tena Dubcek, John Peurifoy, Scott Skirlo, Yann LeCun, Max Tegmark, and Marin Soljačić, "Tunable efficient unitary neural networks (eunn) and their application to rnns," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (JMLR. org, 2017) pp. 1733–1741.

[26] Alexandre Silva, Francesco Monticone, Giuseppe Castaldi, Vincenzo Galdi, Andrea Alù, and Nader Engheta, "Performing Mathematical Operations with Metamaterials," Science **343**, 160–163 (2014).

[27] Michiel Hermans, Michaël Burm, Thomas Van Vaerenbergh, Joni Dambre, and Peter Bienstman, "Trainable hardware for dynamical computing using error backpropagation through physical media," Nature Communications **6**, 6729 (2015).

[28] Cheng Guo, Meng Xiao, Momchil Minkov, Yu Shi, and Shanhui Fan, "Photonic crystal slab Laplace operator for image differentiation," Optica **5**, 251–256 (2018).

[29] Hoyeong Kwon, Dimitrios Sounas, Andrea Cordaro, Albert Polman, and Andrea Alù, "Nonlocal Metasurfaces for Optical Signal Processing," Physical Review Letters **121**, 173004 (2018).

[30] Nasim Mohammadi Estakhri, Brian Edwards, and Nader Engheta, "Inverse-designed metastructures that solve equations," Science **363**, 1333–1338 (2019).

[31] Ardavan F. Oskooi, Lei Zhang, Yehuda Avniel, and Steven G. Johnson, "The failure of perfectly matched layers, and towards their redemption by adiabatic absorbers," Optics Express **16**, 11376–11392 (2008).

[32] William C. Elmore and Mark A. Heald, *Physics of Waves* (Courier Corporation, 2012).

[33] Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980 [cs] (2014).

[34] "`wavetorch`: A Python framework for simulating and back propagating through the wave equation," `https://github.com/fancompute/wavetorch`.

[35] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," Neural computation **9**, 1735–1780 (1997).

[36] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv:1412.3555 (2014).

SUPPLEMENTARY MATERIALS

S1. DERIVATION OF THE WAVE EQUATION UPDATE RELATIONSHIP

In the main text, we specified that the dynamics of the scalar field distribution, $u = u(x, y, z, t)$, are governed by the wave equation

$$\frac{\partial^2 u}{\partial t^2} - c^2 \cdot \nabla^2 u = f, \tag{S1}$$

where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ is the Laplacian operator. $c = c(x, y, z)$ is the spatial distribution of the wave speed and $f = f(x, y, z, t)$ is a source term. For a nonlinear system, c can depend on the wave amplitude. Eq. (S1) may be discretized in time using centered finite differences with a temporal step size of Δt, after which it becomes

$$\frac{u_{t+1} - 2u_t + u_{t-1}}{\Delta t^2} - c^2 \cdot \nabla^2 u_t = f_t. \tag{S2}$$

Here, the subscript in $(\cdot)_t$ is used to indicate the value of a scalar field at time step t. To connect Eq. (S2) to the RNN update equations from Eq. 1 and 2, we exress this in matrix form as

$$\begin{bmatrix} u_{t+1} \\ u_t \end{bmatrix} = \begin{bmatrix} 2 + \Delta t^2 \cdot c^2 \cdot \nabla^2 & -1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_t \\ u_{t-1} \end{bmatrix} + \Delta t^2 \cdot \begin{bmatrix} f_t \\ 0 \end{bmatrix}. \tag{S3}$$

Then, the update equation for the wave equation defined by Eq. (S3) can be rewritten as

$$\boldsymbol{h}_t = \boldsymbol{A}(\boldsymbol{h}_{t-1}) \cdot \boldsymbol{h}_{t-1} + \boldsymbol{P}^{(\mathrm{i})} \cdot \boldsymbol{x}_t \tag{S4}$$

$$\boldsymbol{y}_t = \left| \boldsymbol{P}^{(\mathrm{o})} \cdot \boldsymbol{h}_t \right|^2, \tag{S5}$$

where we have defined $\boldsymbol{A}$ as the matrix appearing in Eq. (S3). The nonlinear dependence on $\boldsymbol{h}_{t-1}$ is defined by the nonlinear wave speed described above.

An absorbing region is introduced to approximate an open boundary condition [31], corresponding to the grey region in Fig. 2B. This region is defined by a dampening coefficient, $b(x, y)$, which has a cubic dependence on the distance from the interior boundary of the layer. The scalar wave equation with damping is defined by the inhomogeneous partial differential equation [32]

$$\frac{\partial^2 u}{\partial t^2} + 2b \cdot \frac{\partial u}{\partial t} = c^2 \cdot \nabla^2 u + f, \tag{S6}$$

where u is the unknown scalar field, b is the dampening coefficient. Here, we assume that b can be spatially varying but is frequency-independent. For a time step indexed by t, Eq. S6 is discretized using *centered* finite differences in time to give

$$\frac{u_{t+1} - 2u_t + u_{t-1}}{\Delta t^2} + 2b\frac{u_{t+1} - u_{t-1}}{2\Delta t} = c^2 \nabla^2 u_t + f_t. \tag{S7}$$

From Eq. S7, we may form a recurrence relation in terms of u_{t+1}, which leads to the following update equation

$$\begin{aligned} \left(\frac{1}{\Delta t^2} + \frac{b}{\Delta t}\right) u_{t+1} - \frac{2}{\Delta t^2} u_t + \left(\frac{1}{\Delta t^2} - \frac{b}{\Delta t}\right) u_{t-1} &= c^2 \cdot \nabla^2 u_t + f_t \\ \left(\frac{1}{\Delta t^2} + \frac{b}{\Delta t}\right) u_{t+1} &= \frac{2}{\Delta t^2} u_t - \left(\frac{1}{\Delta t^2} - \frac{b}{\Delta t}\right) u_{t-1} + c^2 \cdot \nabla^2 u_t + f_t \\ u_{t+1} &= \left(\frac{1}{\Delta t^2} + \frac{b}{\Delta t}\right)^{-1} \left[\frac{2}{\Delta t^2} u_t - \left(\frac{1}{\Delta t^2} - \frac{b}{\Delta t}\right) u_{t-1} + c^2 \cdot \nabla^2 u_t + f_t\right]. \end{aligned} \tag{S8}$$

Equation S8 therefore represents the discretized update equation for the scalar wave equation with damping. In matrix form, we may express Eq. (S8) as

$$\begin{bmatrix} u_{t+1} \\ u_t \end{bmatrix} = \begin{bmatrix} \frac{2 + \Delta t^2 \cdot c^2 \cdot \nabla^2}{1 + \Delta t \cdot b} & \frac{-1 - \Delta t \cdot b}{1 + \Delta t \cdot b} \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_t \\ u_{t-1} \end{bmatrix} + \Delta t^2 \cdot \begin{bmatrix} f_t \\ 0 \end{bmatrix}, \tag{S9}$$

which also has the same form as in Eq. (5) and (6) of the main text.

S2. INPUT AND OUTPUT CONNECTION MATRICES

In the wave equation, we use the linear operators $\boldsymbol{P}^{(\mathrm{i})}$ and $\boldsymbol{P}^{(\mathrm{o})}$ to define the injection and measurement locations within our domain. Here we provide more details on these operators.

We start from the vectors $\boldsymbol{u}_t$ and $\boldsymbol{f}_t$, which are discretized and flattened vectors from the field distribution u_t and f_t. Then, we define the linear operators, $\boldsymbol{M}^{(\mathrm{i})}$ and $\boldsymbol{M}^{(\mathrm{o})}$, each column of which define the respective spatial distributions of the injection and measurement points in this flattened basis. With this, we can write the injection of the input vector ($\boldsymbol{x}_t$) as a matrix-vector multiplication

$$\Delta t^2 \boldsymbol{f}_t \equiv \boldsymbol{M}^{(\mathrm{i})} \cdot \boldsymbol{x}_t. \tag{S10}$$

Similarly, as the output of the RNN at each time step is given by an intensity measurement of the scalar fields, we may express this in terms of the flattened scalar field as

$$\boldsymbol{y}_t = |\boldsymbol{M}^{(\mathrm{o})T} \cdot \boldsymbol{u}_t|^2. \tag{S11}$$

As the wave equation *hidden state*, $\boldsymbol{h}_t$ is defined as the concatenation of $\boldsymbol{u}_t$ and $\boldsymbol{u}_{t-1}$, we define the following matrices for convenience, as they only act on the $\boldsymbol{u}_t$ portion of $\boldsymbol{h}_t$

$$\boldsymbol{P}^{(\mathrm{i})} \equiv \begin{bmatrix} \boldsymbol{M}^{(\mathrm{i})} \\ \mathbf{o} \end{bmatrix} \tag{S12}$$

$$\boldsymbol{P}^{(\mathrm{o})} \equiv [\boldsymbol{M}^{(\mathrm{o})T},\ \mathbf{o}], \tag{S13}$$

where $\mathbf{o}$ is a matrix of all zeros. These matrices are used in the injection and measurement stages of the scalar wave update equations of the main text and thus serve a similar role to the $\boldsymbol{W}^{(x)}$ and $\boldsymbol{W}^{(y)}$ matrices of the traditional RNN in Eqs. (1) and (2). However, unlike $\boldsymbol{W}^{(x)}$ and $\boldsymbol{W}^{(y)}$, these matrices are fixed and not trainable parameters.

S3. TRAINING METHOD FOR VOWEL CLASSIFICATION

The procedure for training the vowel recognition system is as follows. First, each vowel waveform is downsampled from its original recording with a 16 kHz sampling rate to a sampling rate of 10 kHz. Next, the entire dataset of (3 classes) $\times$ (45 males + 48 females) = 272 vowel samples is divided into 5 groups of approximately equal size. Cross validated training is performed with 4 out of the 5 sample groups forming a training set and 1 out of the 5 sample groups forming a testing set. Independent training runs are performed with each of the 5 groups serving as the testing set, with results being averaged over all training runs. Each training run is performed for 30 epochs using the Adam optimization algorithm [33]. During each epoch, every sample vowel sequence from the training set is windowed to a length of 1000, taken from the center of the sequence. This limits the computational cost of the training proceedure by reducing the length of the time through which gradients must be tracked.

All windowed samples from the training set are run through the simulation in batches of 9 and the categorical cross entropy loss between the output probe probability distribution and the correct one-hot vector for each vowel sample is computed. To encourage the optimizer to produce a binarized distribution of the wave speed with relatively large feature sizes, the optimizer minimizes this loss function with respect to a material density distribution, $\rho(x, y)$ within a central region of the simulation domain, indicated by the green region in Fig. 2A. The distribution of the wave speed, $c(x, y)$, is computed by first applying a low-pass spatial filter and then a projection operation to the density distribution. The details of this process are described in section S4. We use the Adam algorithm [33] with learning rate 0.0004 to perform the optimization in batches of 9. Fig. 2D illustrates the optimization process over several epochs, during which, the wave velocity distribution converges to a final structure. At the end of each epoch, the classification accuracy is computed over both the testing and training set. Unlike the training set, the full length of each vowel sample from the testing set is used.

The mean energy spectrum of the three vowel classes after downsampling to 10 kHz is shown in Fig. S1. We observe that the majority of the energy for all vowel classes is below 1 kHz and that there is strong overlap between the mean peak energy of the *ei* and *iy* vowel classes. Moreover, the mean peak energy of the *ae* vowel class is very close to the peak energy of the other two vowels. Therefore, the vowel recognition task learned by the system in the main text is non-trivial.

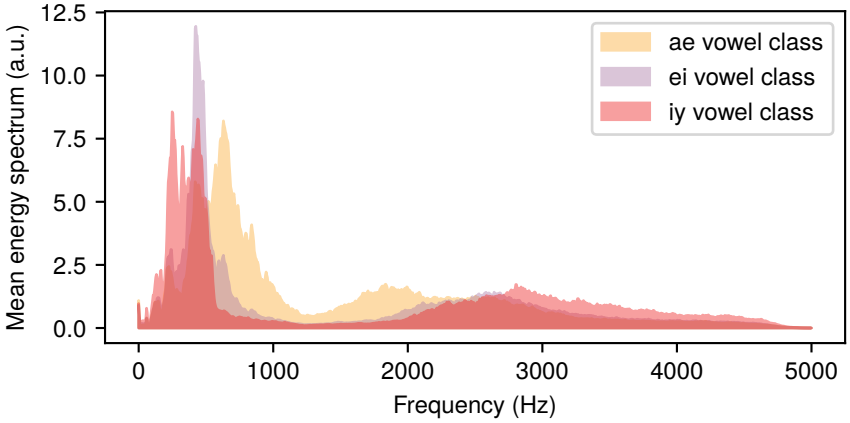

Figure S1. Mean energy spectrum for the *ae*, *ei*, and *iy* vowel classes.

S4. NUMERICAL MODELING METHODS

Numerical modeling and simulation of the wave equation physics was performed using a custom software package written in Python [34]. The software package was developed on top of the popular machine learning library, `pytorch`, to compute the gradients of the loss function with respect to the material distribution via reverse-mode automatic differentiation. In the context of inverse design in the fields of physics and engineering, this method of gradient computation is commonly referred to as the adjoint variable method and has a computational cost of one additional wave simulation. Using of a machine learning platform to perform the numerical simulation greatly reduces opportunities for errors in the analytic derivation or numerical implementation of the gradient. The code for performing numerical simulations and training of the wave equation, as well as generating the figures presented in this paper, may be found online at `http://www.github.com/fancompute/wavetorch/`.

In `wavetorch`, the operation of ∇^2 on a spatially discretized wave field, u_t is carried out using the convolution

$$\nabla^2 u_t = \frac{1}{h^2}\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * u_t, \tag{S14}$$

where h is the step size of the spatial grid and $*$ denotes a convolution.

To create realistic devices with sufficiently large minimum feature sizes and a binarized $c(x, y)$ distribution, we employed filtering and projection schemes during our optimization. Rather than updating the wave speed distribution directly, one may instead choose to update a design density $\rho(x, y)$, which varies between 0 and 1 within the design region and describes the *density* of material in each pixel. To create a structure with larger feature sizes, a low pass spatial filter can be applied to $\rho(x, y)$ to created a filtered density, labelled $\tilde{\rho}(x, y)$

$$\tilde{\rho}(x, y) = \begin{bmatrix} 0 & 1/8 & 0 \\ 1/8 & 1/2 & 1/8 \\ 0 & 1/8 & 0 \end{bmatrix} * \rho(x, y). \tag{S15}$$

For binarization of the structure, a projection scheme is used to recreate the final wave speed from the filtered density. We define $\bar{\rho}(x, y)$ as the projected density, which is created from $\tilde{\rho}(x, y)$ as

$$\bar{\rho}_i = \frac{\tanh(\beta\eta) + \tanh(\beta[\tilde{\rho}_i - \eta])}{\tanh(\beta\eta) + \tanh(\beta[1 - \eta])}. \tag{S16}$$

Here, η is a parameter between 0 and 1 that controls the mid-point of the projection, typically 0.5, and β controls the strength of the projection, typically around 100.

Finally, the wave speed can be determined from $\bar{\boldsymbol{\rho}}$ as

$$c(x, y) = (c_1(x, y) - c_0(x, y))\bar{\boldsymbol{\rho}} + c_0(x, y), \tag{S17}$$

where c_0 and c_1 are the background and optimized material wave speed, respectively.

Model	Nonlinearity	Number of parameters	Accuracy	
			Training	Testing
Wave Equation	linear wave speed	4200	93.1%	86.6%
	nonlinear wave speed	4200	92.6%	86.3%
Conventional RNN	linear	5250	78.8%	79.4%
	leaky ReLU	5250	82.6%	80.2%
	linear	10500	88.9%	88.2%
	leaky ReLU	10500	89.4%	89.4%

Table S1. Comparison of scalar wave model and conventional RNN on vowel recognition task.

S5. COMPARISON OF THE WAVE EQUATION AND A CONVENTIONAL RNN

We compare the wave equation results to a conventional RNN model as defined in Eq. (1) and Eq. (2). The number of trainable parameters in the model is determined by the hidden state size N_h, as the model is given by three matrices $\boldsymbol{W}^{(x)}, \boldsymbol{W}^{(h)}$, and $\boldsymbol{W}^{(y)}$ of size $[N_h, 1]$, $[N_h, N_h]$ and $[3, N_h]$, respectively. We tried N_h =70, for which the total number of RNN free parameters is 5250, and N_h =100, with 10500 free parameters. The RNN was implemented and trained using `pytorch`. In Table S1 we show the results of a standard RNN on the vowel recognition task and compare them to the scalar wave. We find that the conventional RNN achieves a performance comparable to the wave equation. However, this performance is highly dependent on the number of trainable parameters. For a similar number of trainable parameters, the conventional RNN achieves about 6% lower classification accuracy. However, when the number of free parameters is increased to about 2 times that of the scalar wave, the accuracy is higher by about 3 %. We note that it is possible that more advanced recurrent models like long short-term memory (LSTM) [35] or gated recurrent unit (GRU) [36] could have a better performance with a smaller number of parameters, but exploring this is outside the scope of this study.

The conventional RNN and the one implemented by a scalar wave equation have many qualitative differences. We discuss some of those below. First, in the RNN case, the trainable parameters are given by the elements of the weight matrices. In the wave equation case, we choose to use the wave velocity, $c(x, y, z)$, as trainable parameters, because a specific distribution of c can be physically implemented after the training process. In acoustic or optical systems, this can be practically realized using technologies such as 3D printing or nanolithography. Furthermore, whereas the RNN free parameters define a matrix which is multiplied by the input, output, and hidden state vectors, in the wave equation case, the free parameters are multiplied element-wise with the hidden state, which limits the influence of each individual parameter over the full dynamics.

For a given amount of expressive power, the size of the hidden state in the wave equation must arguably be much larger than that in the RNN case. This is because the amount of information that can be encoded in the spatial distribution of u_t is constrained by the diffraction limit for wave systems. It follows that a single RNN hidden state element may be analogous to several grid cells in the scalar wave equation. Furthermore, the wave update matrix A is sparse and only contains non-zeros on diagonal elements (self coupling) and those corresponding to neighbor-to-neighbor coupling between spatial grid cells. Because of this, information in a given cell of u_t will take several time steps to reach other cells, as determined by the wave velocity and the distance between them. The presence of this form of causality practically means that one must wait longer for a full 'mixing' of information between cells in the domain, suggesting that in the our numerical simulations, a larger number of time steps may be needed as compared to the typical RNN.

The form of nonlinearity used in the wave equation is different from that in the typical RNN, which involves the application of the nonlinear function, $\sigma^{(h)}(\cdot)$, as in Eq. (1). In the wave equation, nonlinearity is provided by making the wave velocity, c, or damping dependent, b, to be dependent on the instantaneous wave intensity $|u_t|^2$. For example $c = c(|u_t|^2)$, or $b = b(|u_t|^2)$. In optics, these nonlinearities may be implemented using $\chi^{(3)}$ materials or saturable absorption, respectively. With this addition, the update matrix of Eq. (5), $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{h}_{t-1})$, becomes a function of the solution at that time step, making the dynamics nonlinear. Nonlinearity is introduced into the output of the wave system ($\boldsymbol{y}_t$) by measuring the intensity of the wave field, which involves a squaring operation. One may also consider directly discretizing a nonlinear wave equation using the same technique as the main text.