

Supplementary Information for:

Reconfigurable Versatile Integrated Photonic Computing Chip

Yufei Wang^{1,2†}, Kun Liao^{1†*}, Kuo Zhang^{3†}, Zhuochen Du^{1†}, Ze Wang^{1†}, Bo Ni^{1,2}, Tianyu Xu¹, Shuai Feng³, Yan Yang^{4*}, Qi-Fan Yang^{1,2,5,6*}, Quan Sun^{2*}, Xiaoyong Hu^{1,2,5,6*}, Qihuang Gong^{1,2,5,6}

¹State Key Laboratory for Mesoscopic Physics & Department of Physics, Collaborative Innovation Center of Quantum Matter & Frontiers Science Center for Nano-optoelectronics, Peking University, Beijing 100871, China

²Key Laboratory for Advanced Optoelectronic Integrated Chips of Jiangsu Province, Peking University Yangtze Delta Institute of Optoelectronics, Nantong, Jiangsu 226010, China

³School of Science, Minzu University of China, Beijing 100081, China

⁴Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China

⁵Collaborative Innovation Center of Extreme Optics, Shanxi University, Taiyuan, Shanxi 030006, China

⁶Hefei National Laboratory, Hefei 230088, China

† These authors contributed equally to this work.

*E-mail: kunliao@pku.edu.cn; yyang10@ime.ac.cn; leonardoyoung@pku.edu.cn; sunquan@ydioe.pku.edu.cn; xiaoyonghu@pku.edu.cn

Supplementary Notes:

Supplementary Note 1 | Reconfigurable Versatile Integrated Photonic Computing Chip based on the unified structure.

Supplementary Note 2 | Principle of the Fully Connected Neural Network with negative weight and input.

Supplementary Note 3 | Principle of the Convolutional Neural Network with various kernel size.

Supplementary Note 4 | Principle of the Photonic Gated Recurrent Neural Network.

Supplementary Note 5 | Experimental setup.

Supplementary Note 6 | Integrated turnkey soliton microcomb generation.

Supplementary Note 7 | Calibration of input encoding and weight deployment.

Supplementary Note 8 | Supplementary training process and sample results for sentiment analysis tasks.

Supplementary Note 9 | FCNN-CNN-PGRNN network schematic and detailed completion process of speech recognition tasks.

Supplementary Note 10 | Area/Power Efficiency and Computing Accuracy of the Photonic Computing Chip.

Supplementary Note 11 | Scalability Potential of the Photonic Computing Chip.

Supplementary Note 12 | Comparison with the state-of-the-art photonic neural networks.

Supplementary Figures:

Fig. S1 | Comparison between single, hybrid and unified configurations.

Fig. S2 | Principle of the CNN with 1×1 kernel size.

Fig. S3 | Principle of the CNN with 2×2 kernel size.

Fig. S4 | Principle of the CNN with 4×4 kernel size.

Fig. S5 | Principle of the CNN with 8×8 kernel size.

Fig. S6 | Schematic of cross-waveguide based MRR.

Fig. S7 | Electric tuning model of MRR based on cross-waveguide coupling.

Fig. S8 | Schematic illustration of using MRRs to implement RNN operations.

Fig. S9 | Comparison between RNN and PGRNN.

Fig. S10 | Experimental setup for single-sided input.

Fig. S11 | The experimental setup including all integrated and non-integrated components.

Fig. S12 | Integrated turnkey soliton microcomb.

Fig. S13 | The power variation curve of the 8-channel comb teeth under modulation.

Fig. S14 | The normalized calibration power variation curve of the 8-channel comb teeth under modulation.

Fig. S15 | Demonstration of the fine-tuned voltage control process.

Fig. S16 | Output power curve of MRR modulated by voltage (before unified).

Fig. S17 | Output power curve of MRR modulated by voltage (after unified).

Fig. S18 | Normalized power output of the MRR under different inputs.

Fig. S19 | Normalized weight of the MZI under modulation.

Fig. S20 | Vocabulary Construction.

Fig. S21 | Physical architecture of the photonic chip used for speech recognition.

Fig. S22 | Layer-wise execution flow and data movement throughout the system.

Fig. S23 | Sample cases from the LJ-speech dataset.

Fig. S24 | Loss and WER on the LJ-Speech dataset.

Fig. S25 | Speech recognition tasks.

Fig. S26 | Comparison between the target and predicted texts.

Fig. S27 | Impact of thermal drift on computation.

Fig. S28 | Comparison with the state-of-the-art photonic neural networks in the tunable area efficiency and neurons.

Supplementary Table:

Table. S1 | Comparison of different source type in the key performance.

Table. S2 | Sample cases from the IMDB dataset.

Table. S3 | Energy consumption breakdown of the photonic computing chip.

Table. S4 | Comparison with the state-of-the-art photonic neural networks in the key performance.

Supplementary Note 1 | Reconfigurable Versatile Integrated Photonic Computing Chip based on the unified structure.

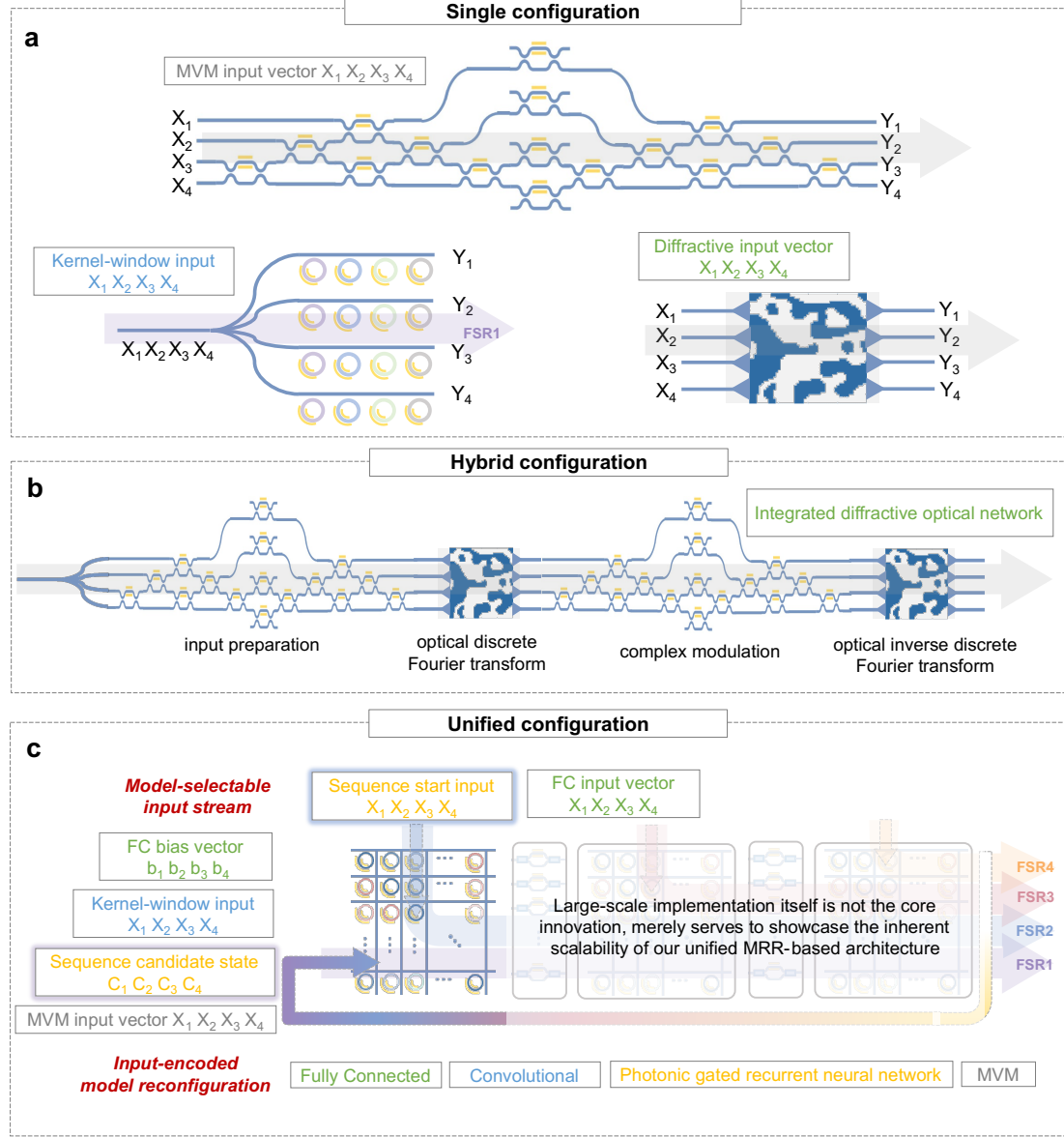


Fig. S1 | Comparison between single, hybrid and unified configurations.

To further explain why our unified configuration uniquely accommodates dynamic temporal models, we present Fig. S1. The key distinction between static and dynamic tasks lies in temporal dependency: dynamic tasks require the network to simultaneously process information from multiple time steps, whereas static tasks involve single-pass inference with no dependence on prior inputs. Fig. S1 illustrates three representative hardware configurations used in photonic neural computing.

In the single configuration (Fig. S1a), signal propagation is typically unidirectional—from left to right—mapping an input vector to an output vector in a static manner. Such structures are commonly adapted to represent dense layers, convolution kernels, or other fixed-function mappings, and are generally well-suited for feedforward inference on static tasks. However, they inherently lack the architectural flexibility to simultaneously process multiple time steps, as required in dynamic, sequential tasks. Because data flows in only one direction, there is no mechanism for integrating temporal context from previous inputs during inference.

In the hybrid architecture (Fig. S1b), system performance is improved by functionally distributing different tasks, e.g., enhance energy efficiency, scalability, or latency, but still retain the unidirectional, single-pass data flow typical of static models. As such, they likewise struggle to support tasks requiring joint reasoning over multiple time steps. While some prior works, such as “Nat Commun 15, 10841 (2024)”, they have rearchitected their systems specifically for dynamic tasks, such designs remain relatively uncommon and often require significant additional hardware complexity.

By contrast, our unified configuration (Fig. S1c) naturally supports time-sequential inference. This is achieved through the dual-port access of our core computing elements: each MRR unit can simultaneously respond to two distinct input streams, representing, for example, the current input and the hidden state from a previous time step. This enables the network to merge past and present information in real time, closely emulating the behavior of electronic recurrent networks. Importantly, this capability arises not from layering separate components, but from a coherent design of the photonic architecture and its usage protocol, resulting in both hardware efficiency and model diversity.

Supplementary Note 2 | Principle of the Fully Connected Neural Network with negative weight and input.

As shown in Fig. 1b, we use wavelengths λ_l - λ_m resonating with the MRRs, to carry input information X_l - X_m . These wavelengths enter from the upper ports of the crossing

waveguide. The inputs corresponding to the first column of N MRRs are all X_1 , and similarly, the inputs corresponding to the M -th column of n MRRs are all X_m . The output ports on the right side correspond to their respective drop ports. The sum of the optical power detected from the output ports across all wavelengths gives the result:

$$Y_n = W_{n1}X_1 + W_{n2}X_2 + \dots + W_{nm}X_m. \quad (1)$$

When inputting the X information at the upper port, we input a bias b_1 - b_n at the detuned wavelength λ_{m+1} from the left port of the crossing waveguide. Since the optical signal from the left port is detuned relative to the MRRs, it is not modulated, which aligns with the concept of bias. Thus, at the output port, we can detect the sum of the weighted input and the bias:

$$Y_n = W_{n1}X_1 + W_{n2}X_2 + \dots + W_{nm}X_m + b_n. \quad (2)$$

It is important to note that the crossing waveguide not only connects the horizontal MRRs but also connects the vertical MRRs. Therefore, the wavelengths of the vertical N MRRs and horizontal M MRRs must be staggered. For example, in an 8×8 MRR array, the first row's MRR resonance frequencies need to be tuned to $\lambda_1, \lambda_2, \dots, \lambda_8$, the second row to $\lambda_2, \lambda_3, \dots, \lambda_8, \lambda_1$, the third row to $\lambda_3, \lambda_4, \dots, \lambda_8, \lambda_1, \lambda_2$, and the eighth row to $\lambda_8, \lambda_1, \dots, \lambda_7$. This ensures that there is no crosstalk between the vertical and horizontal MRRs. Therefore, the formula is further written as:

$$Y_n = [W_{n1}X_1]_{\lambda(m-n+2) \bmod(m)} + [W_{n2}X_2]_{\lambda(m-n+3) \bmod(m)} + \dots + [W_{nm}X_m]_{\lambda(m-n+m+1) \bmod(m)} + [b_n]_{\lambda(m+1)}. \quad (3)$$

Although a single MRR can only perform positive weight operations in a single computation, when we need to perform weight operations on real numbers that include both positive and negative values, we can achieve this by sacrificing the number of operations. An offset matrix $\mathbf{W}_{\text{offset}}$ was introduced to the computing processing. By adding this to the original weight matrix $\mathbf{W}$, we obtain a new matrix $\mathbf{W}'$ with non-negative elements. After calculating $\mathbf{W}'$, we computing separately on $\mathbf{W}'$ and $\mathbf{W}_{\text{offset}}$ to get $\mathbf{W}'\mathbf{X}$ and $\mathbf{W}_{\text{offset}}\mathbf{X}$, respectively. Since all elements of $\mathbf{W}_{\text{offset}}$ are identical, the size of the MRR crossbar array for computing $\mathbf{W}_{\text{offset}}$ can be reduced. The original $\mathbf{WX}$ can then be derived using $2(\mathbf{W}'\mathbf{X} - \mathbf{W}_{\text{offset}}\mathbf{X})$. Regarding computational complexity, for the matrix-vector multiplication operation, if $\mathbf{W}$ is of size $N \times N$, an

$N \times N$ MRR crossbar array is needed for $\mathbf{W}'\mathbf{X}$. In contrast, only a $1 \times N$ MRR crossbar array is required for $\mathbf{W}_{\text{offset}}\mathbf{X}$. Overall, an $N \times (N + 1)$ MRR crossbar array is necessary for MVM including negative values. Since the MRR needed for $\mathbf{W}_{\text{offset}}\mathbf{X}$ does not scale with the square of N , the impact of $\mathbf{W}_{\text{offset}}\mathbf{X}$ on computational complexity is negligible when N is large. To accomplish a computation with a negative input, the operational principle is similar to that of negative weights, both can be achieved by slightly increasing the computational load.

Supplementary Note 3 | Principle of the Convolutional Neural Network with various kernel size.

Fig. 1c illustrates the computation scheme for a 1×1 convolution kernel. For the MRR array, it is capable of performing convolution operations of varying kernel sizes and multi-channel to multi-channel convolutions through flexible configuration of input encoding and weight deployment. As we mentioned in the methods section of the main text, for a 1×1 convolution kernel, the input is $X_l - X_m$ in each row. However, this is not the case for larger convolution kernels. As shown in Figs. S2-5, which depicts the computation schemes for 1×1 , 2×2 , 4×4 and 8×8 convolution kernels.

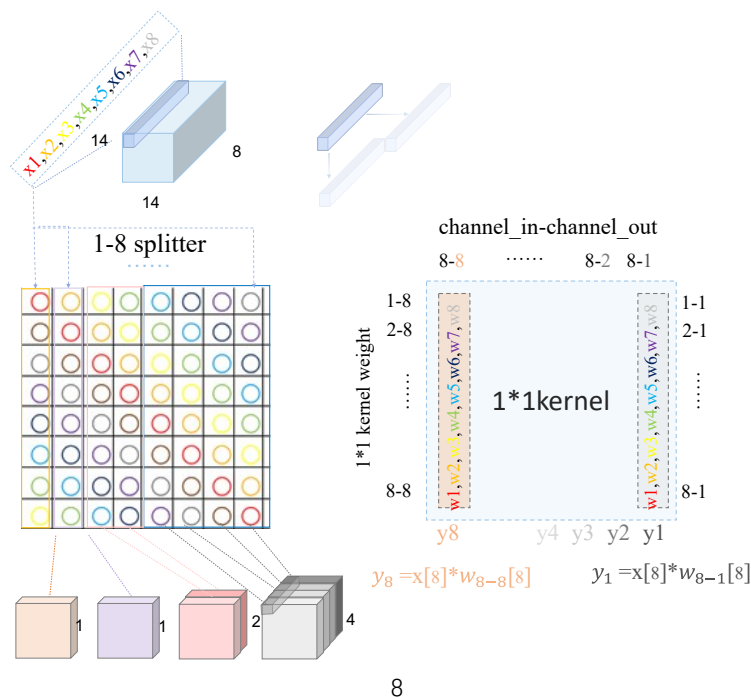


Fig. S2 | Principle of the CNN with 1×1 kernel size.

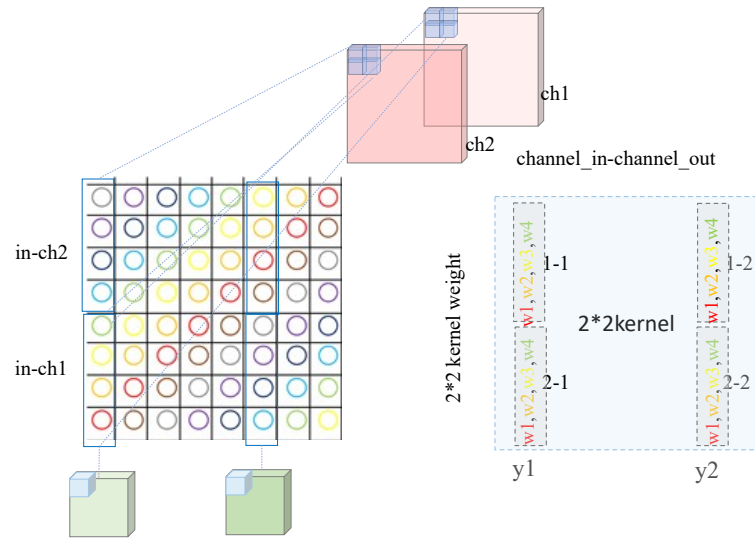


Fig. S3 | Principle of the CNN with 2×2 kernel size.

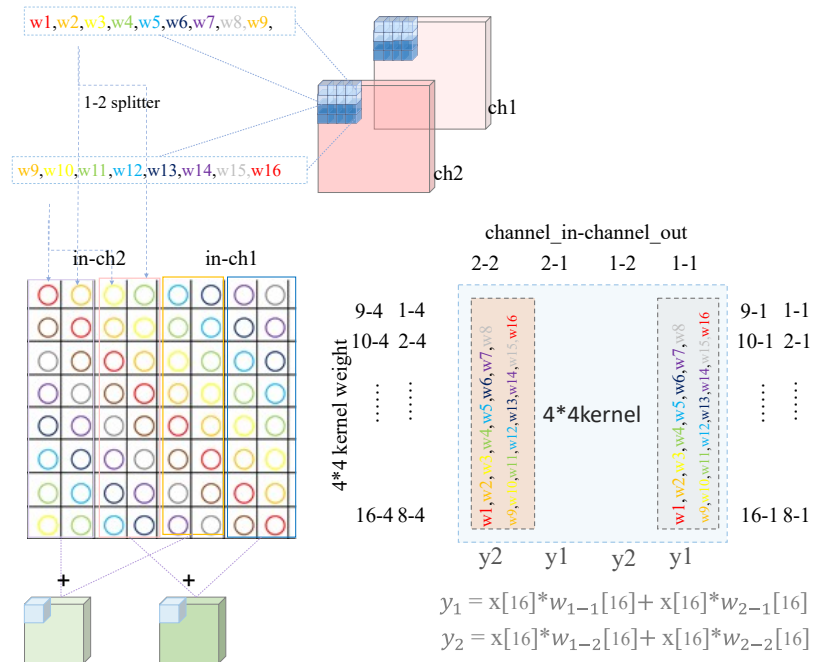


Fig. S4 | Principle of the CNN with 4×4 kernel size.

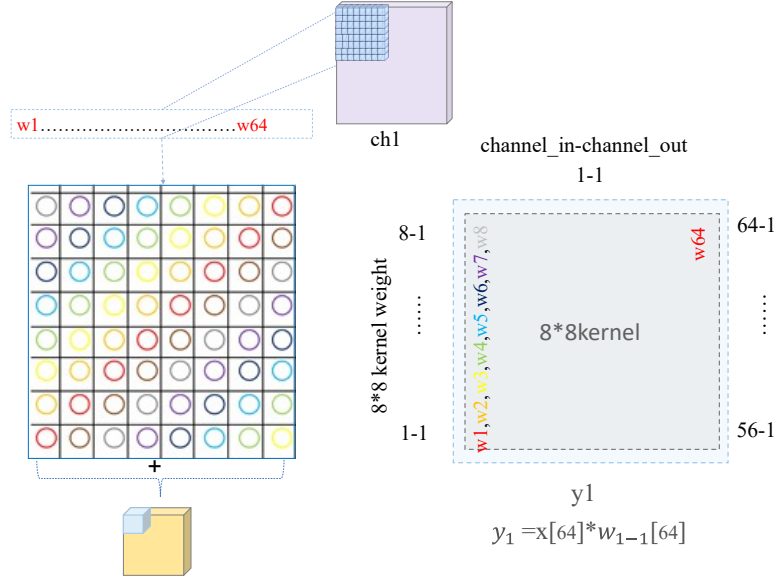


Fig. S5 | Principle of the CNN with 8×8 kernel size.

Supplementary Note 4 | Principle of the Photonic Gated Recurrent Neural Network.

The notable feature of RNNs is their ability to capture temporal dependencies in sequential data, which is attributed to their unique network structure. Here, we employ a cross-waveguide MRR as a single neuron in the PGRNN. Specifically, each neuron receives two primary inputs: one is the input data at the current time step, and the other is the hidden state passed down from the previous time step. These two inputs jointly determine the output of the current neuron, which is the hidden state at the current time step. The hidden state contains both the information of the current time step and the memory of the sequence context, which is recursively passed to the next time step.

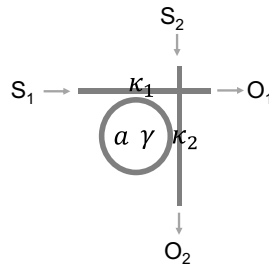


Fig. S6 | Schematic of cross-waveguide based MRR.

Fig. S6 delineates the sections where energy exchange occurs between the

microcavity and the waveguides, which can be primarily divided into the following parts: the incident light from the waveguides (with S_1 and S_2 representing the incident lights to the microcavity from the horizontal and vertical waveguides, respectively), the transmitted light from the waveguides (with O_1 and O_2 representing the transmitted lights from the horizontal and vertical waveguides, respectively), the light coupled from the waveguides to the microcavity (with the coupling coefficients κ_1 and κ_2 for the microcavity with the horizontal and vertical waveguides, respectively), the intrinsic loss of the microcavity denoted by γ , and the amplitude transmission of the ring waveguide denoted by a . According to the coupled-mode theory of the microcavity, we can derive the steady-state conditions as follows:

$$\frac{da}{dt} = -i(\omega - \omega_0)a - (\gamma + \kappa_1 + \kappa_2)a - \sqrt{2\kappa_1}S_1 - \sqrt{2\kappa_2}S_2 = 0, \quad (4)$$

solving the equations based on the coupled-mode theory, we can obtain the following:

$$a = \frac{-\sqrt{2\kappa_1}S_1 - \sqrt{2\kappa_2}S_2}{i(\omega - \omega_0) + (\gamma + \kappa_1 + \kappa_2)}. \quad (5)$$

The expression for the output at the horizontal waveguide port, which we use as the output of the MRR neuron, can be formulated as follows:

$$O_1 = S_1 + \sqrt{2\kappa_1}a = S_1 - \frac{2\kappa_1S_1 + 2\sqrt{\kappa_1\kappa_2}S_2}{i(\omega - \omega_0) + (\gamma + \kappa_1 + \kappa_2)}. \quad (6)$$

There are two methods for a single MRR neuron to simultaneously modulate two sets of input signals: the first is to encode the optical signals at both input ports using the same wavelength that resonates with the MRR. We assume the following:

$$S_2 = AS_1e^{i\phi}. \quad (7)$$

Furthermore, due to the identical coupling coefficients between the microcavity with both the vertical and horizontal waveguides, $\kappa_2 = \kappa_1$, O_1 can be expressed as:

$$O_1 = S_1 - \frac{2\kappa_1S_1 + 2\kappa_1AS_1e^{i\phi}}{i(\omega - \omega_0) + (\gamma + \kappa_1 + \kappa_2)}. \quad (8)$$

It can be observed that O_1 depends not only on the intensity of the input but also on the phase. We applied this transmission model to the custom forward propagation model of

a neural network, simulated the optical frequency comb light source, and defined the network training parameters as the shifts of resonant frequency. Theoretically, we demonstrate that this approach can still fulfill the functions of a PGRNN and perform well in sentiment analysis tasks. However, when considering experimental conditions, due to this method imposes strict phase requirements on the input optical signals, necessitating the integration of two additional phase modulators in each MRR to ensure that the initial phase of light entering the same MRR is controllable, which would introduce more complex control and calibration requirements for actual experiments. Therefore, we adopt the second approach, which is a non-coherent computation mechanism. By leveraging the different free spectral ranges (FSRs) of the MRR, input signals at wavelengths λ and $\lambda + \text{FSR}$ are input from the two ports of the cross-waveguide respectively, thus avoiding the influence of phase differences between the two input ports on the output. When the same MRR modulates input signals at wavelengths λ_1 and $\lambda_1 + \text{FSR}_{\text{mrr}}$, the outputs exhibit opposite trends, with the through output decreasing while the drop output increasing.

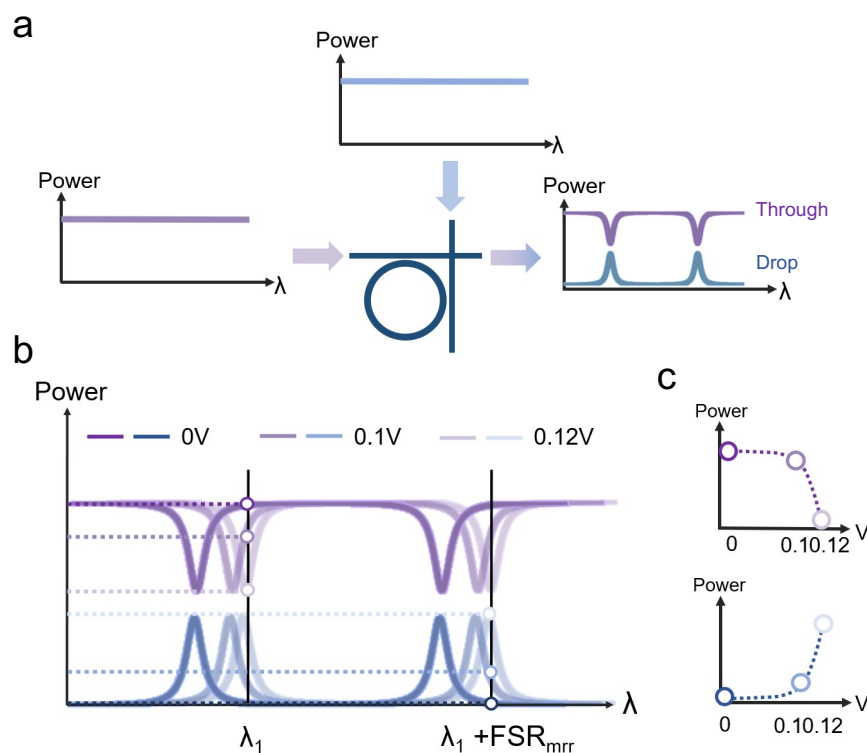


Fig. S7 | Electric tuning model of MRR based on cross-waveguide coupling. (a) MRR input-output model based on cross-waveguide coupling. (b) Schematic of the

voltage-controlled tuning curve of the MRR transmission spectrum. (c) Schematic of the voltage-dependent output curve of the MRR when using an optical frequency comb as the input source.

To clarify how the opposite trends evolve from the spectral domain to the voltage domain, we provide an illustrative example in Fig. S7b. Here, for simplicity and clarity, the linewidth, FSR, transmission levels, and voltage ranges are scaled. Initially, when employing the broadband supercontinuum source, the through and drop spectra (plotted in the same axes for ease of comparison) shift equally in wavelength as the voltage increases from 0 V to 0.12 V. Switching to a comb source effectively selects specific points at λ_1 and $\lambda_1 + \text{FSR}_{\text{mrr}}$ from the spectra, as marked by circles in Fig. S7b. Consequently, on the through port, as the voltage increases, the transmission decreases from the maximum on the right shoulder of a resonance dip toward its minimum at the resonance center. In contrast, on the drop port, transmission increases from the minimum on the resonance peak's shoulder to the maximum at resonance center. Transforming these spectral behaviors into transmission-vs-voltage curves produces two trends with opposite slopes, as depicted schematically in Fig. S7c and experimentally confirmed in Fig. 2j.

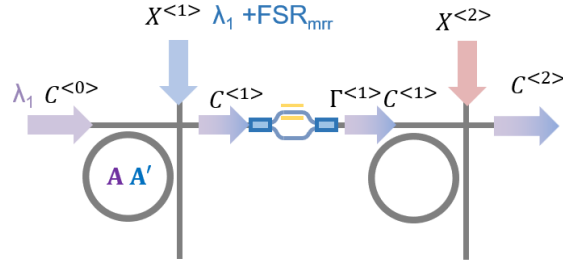


Fig. S8 | Schematic illustration of using MRRs to implement RNN operations. A conceptual diagram showing the correspondence between the input/output of an RNN and that of MRR arrays, using two MRRs as representatives of two layers.

As shown in Fig. S8a, each MRR neuron simultaneously receives and processes the current input (top port) and previous state (left port), enabling the temporal dynamics required by RNNs. The ability of each MRR to modulate two distinct input

signals concurrently, unlike traditional MRR-based computations.

Traditional RNNs on digital computers feature the concept of weight sharing, meaning that the same set of weights is applied across all time steps. However, in our PGRNN, the parameters across the three layers are actually distinct. This is due to the fact that after completing an operation in a time step in a traditional RNN, which includes matrix-vector multiplication between the current layer's input $\mathbf{x}^{<t>}$ and weights $\mathbf{A}$, as well as between the previous layer's output $\mathbf{c}^{<t-1>}$ and weights $\mathbf{A}'$, a new vector is obtained. This vector represents the sum of the outputs from different inputs of the previous layer after being modulated by the weights (the summation is key). Thus, as shown in Fig. S9, the vector regulated by A'_1 in the second layer includes the sum of diverse information such as $x_1^t, x_2^t, c_1^{t-1}, c_2^{t-1}, A_1, A_2, A'_1, A'_2$. The mechanism of the MRR makes the regulation in the second layer more akin to a Hadamard operator, as the MRR only acts on information carried by the resonant frequencies. Therefore, the wavelengths regulated by A'_1 in the second layer can only affect the outputs of neurons that were modulated by the same wavelengths in the previous layer, that is, the outputs x_1^t and c_1^{t-1} modulated by A_1 and A'_1 . To prevent A_1 from continuously stacking effects with A'_1 , we altered the value of A'_1 in the second layer to B'_1 in order to differentiate it from the A'_1 in the first layer, enhancing the diversity of modulation control. The distinct weights of $\mathbf{A}'$ and $\mathbf{A}$ are achieved through the different transmission rates at the drop port and through port of the microcavity. In terms of performance on dynamic temporal tasks, our PGRNN designed for optical structures, can match the capabilities of traditional RNNs on computers.

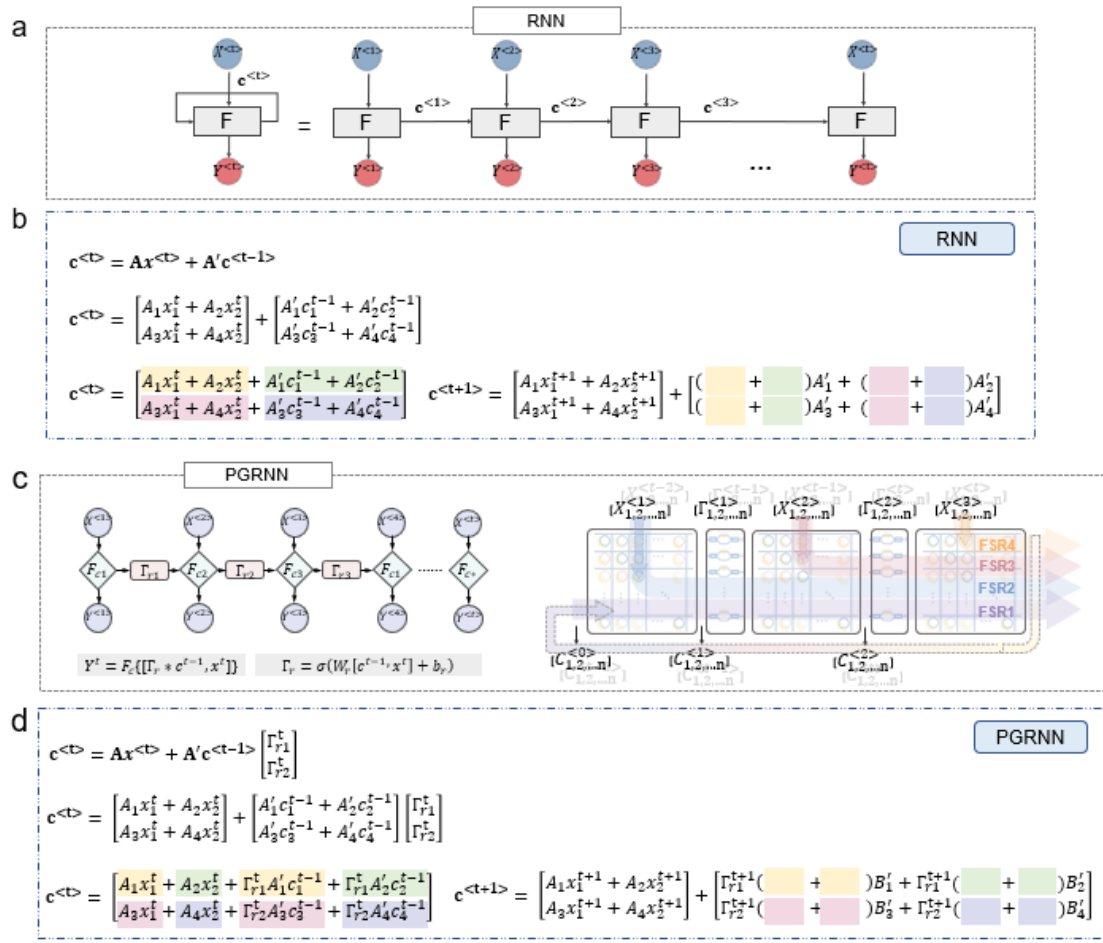


Fig. S9| Comparison between RNN and PGRNN. (a-b) illustrate the flowchart and core equations of the RNN. (c-d) present those of the PGRNN. The blocks with color labels in the c^{t+1} represent the formulas corresponding to the same color in c^t , providing an intuitive view of which parameters from the previous layer are influenced by the weights in the second layer.

Supplementary Note 5 | Experimental setup.

In the experimental setup, an EDFA was placed after the microcomb to compensate for power loss introduced by subsequent modulators. Depending on the computational task, different configurations of DWDM, VOA, and beam splitters were employed. Fig. S10 illustrates the implementation of a CNN using our chip, where the MRR array operated with a single-sided input. In this setup, the consistency of inputs across different rows was determined by the convolutional kernel size. When the inputs were uniform, a single set of DWDM and VOA was used for input encoding, with a 50:50

beam splitter evenly distributing the optical signals into eight distinct ports. Conversely, when input values varied across rows, multiple sets of DWDM and VOA were required to generate the corresponding distinct inputs. Fig. S11 depicts the dual-sided input scenario, where light was injected simultaneously from both ports of the MRR array. This configuration corresponded to FCNN and PGRNN implementations. Similar to the second case in Fig. S10, multiple sets of DWDM and VOA were needed to prepare the inputs. For the output, a multi-channel power meter was used to measure the power across all channels. The computer terminal was responsible for reading the output, controlling the input, and setting the weights for the neural network.

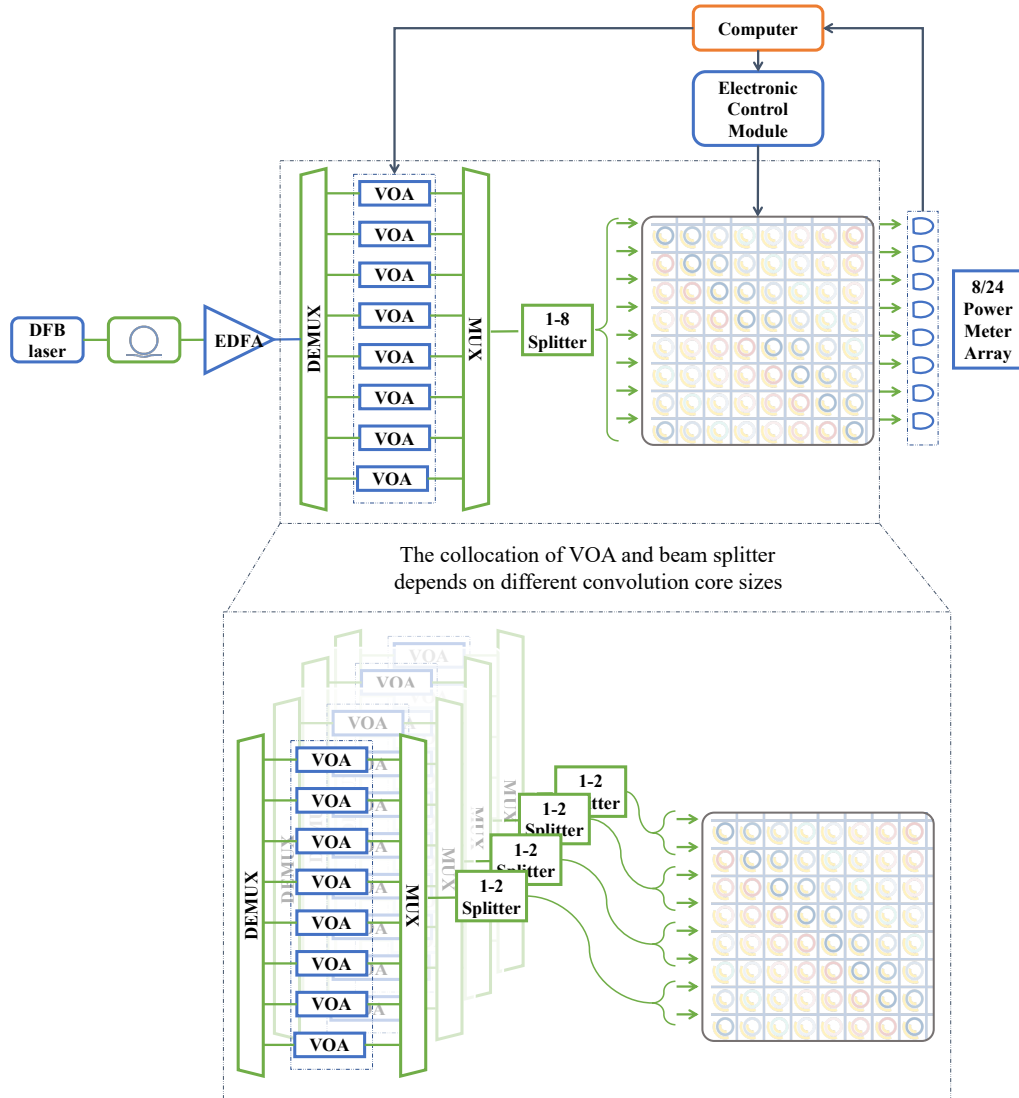


Fig. S10 | Experimental setup for single-sided input.

As shown in Fig. S11, it clearly labels and distinguishes between:

- ① Integrated components, which including two independent modules:
 - The DFB-integrated single soliton microcomb source, used to generate stable multi-wavelength optical carriers.
 - Our custom-designed photonic computing chip, which implements the core neural computing functions using the unified structures.
- ② Non-integrated components:
 - Encoding and modulation hardware: including an EDFA, VOA, DWDM and fiber splitters.
 - Control electronics: such as a CPU and DAC-based drivers used to thermally program the on-chip weights.
 - Detection units: precision power meters to measure output intensities.

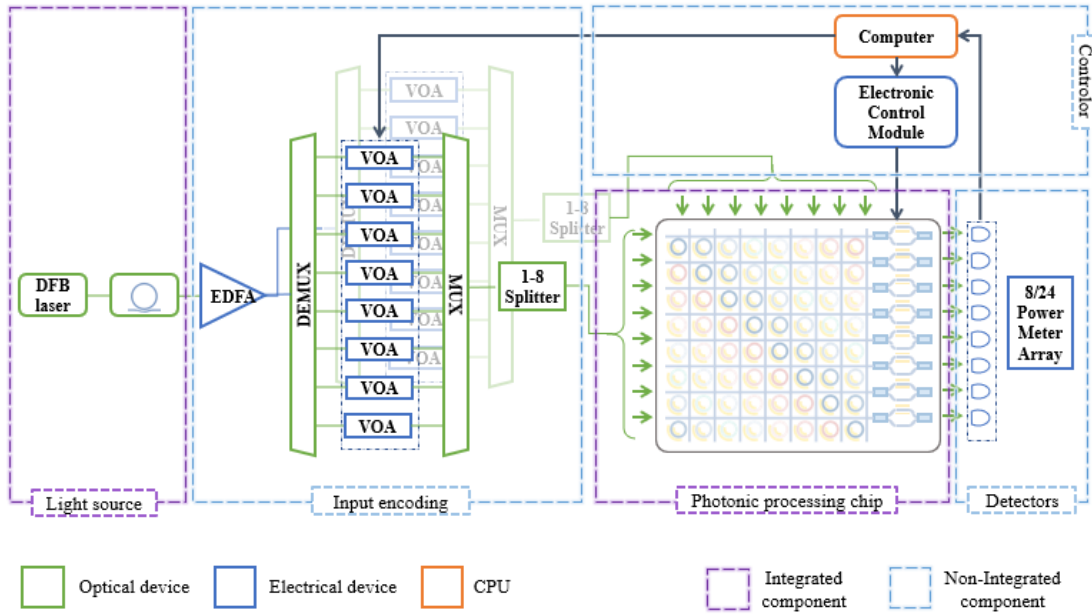


Fig. S11 | The experimental setup including all integrated and non-integrated components.

Each component in the figure is color-coded by its integration status, and optical/electrical signal paths are clearly indicated. We hope this updated diagram now enables readers to better understand the experimental architecture

Looking forward, since the core innovation of our work lies in demonstrating the functional versatility and configurability of the on-chip photonic computing architecture, we did not prioritize full system integration in this initial demonstration. Nevertheless, we fully agree that integration is a central and necessary direction for advancing toward practical deployment, and we are actively exploring strategies to improve it. Our current experimental setup still relies on several discrete fiber-based or benchtop components for signal preparation, modulation, and readout. However, there is significant potential to replace or integrate many of these modules with photonic and electronic integrated components. Specifically:

- ① On-chip integration of optical amplifiers: The EDFA currently used to boost the microcomb output can be replaced by SOA. With recent advances in hybrid III–V/silicon integration, SOA can be monolithically or heterogeneously integrated with the computational photonic chip. This would significantly reduce power consumption, coupling losses, and system footprint (IEEE Journal of Selected Topics in Quantum Electronics, 22(6), 78-88).
- ② Monolithic or hybrid co-integration of photonic and electronic components: Components such as VOA, DWDM, and PD now implemented as external fiber-based devices, which can be fully integrated onto a silicon photonic platform. Mature techniques such as grating coupler-based DWDMs and germanium PDs have already demonstrated feasibility, and commercial platforms are increasingly supporting co-integration of these elements (Nature photonics, 4(8), 527-534.; Nature, 556(7701), 349-354.).

In summary, while our present system adopts a partially hybrid configuration of photonic and electronic components, we fully recognize the importance of advancing toward highly integrated, low-power, and scalable photonic computing architectures. These improvements are a natural next step for translating our reconfigurable photonic neural network platform into practical applications.

Supplementary Note 6 | Integrated turnkey soliton microcomb generation.

The soliton frequency comb is not included merely for demonstrating integration, but is in fact a critical component that directly enables large-scale, parallel, and high-fidelity photonic computing using MRR arrays. Each MRR unit functions as a tunable weight filter, operating at a specific resonant wavelength. To perform multiplication in parallel, each MRR must be addressed by a distinct, stable, and narrowband optical carrier. This requires a broadband, multi-wavelength source with: ① Many stable, evenly spaced lines (for addressing many MRRs simultaneously). ② Narrow linewidths (to reduce channel interference). ③ Mutual coherence (to preserve phase and signal integrity). ④ High per-line power (to ensure sufficient SNR after losses). In this context, a soliton microcomb uniquely satisfies all these demands. It offers hundreds of equidistant, mutually coherent, and ultra-narrow-linewidth spectral lines in a compact and chip-compatible format.

In our architecture, MRRs are densely cascaded, and each ring must respond to a unique frequency. If spectral lines are too wide, they will overlap or require excessive spacing, limiting the number of addressable channels and increasing the device footprint. Narrow linewidths reduce the required channel spacing, increasing the number of MRRs that can be addressed in a given band. Only comb-based sources provide both dense, stable channels and chip-level integration. In practice, whenever researchers have access to an integrated comb, they adopt it for MRR computing. In contrast, using a supercontinuum source combined with wavelength division multiplexers (SC + WDM) is often a compromise made due to limited access to integrated combs, but it comes at the cost of OSNR, channel density, and spectral precision.

To clarify the role and advantages of different light sources in MRR-based photonic computing, we summarize in the following table a comparison among common options including CW lasers, supercontinuum sources with WDM filtering, and soliton frequency combs

Table. S1 | Comparison of different source type in the key performance.

Source type	Channel	Linewidth/	Power per	Suitability for
-------------	---------	------------	-----------	-----------------

	count	mutual coherence	line	large MRR arrays
Tunable CW laser	1	kHz–MHz, coherent	High	× 1 channel
SC + DWDM	~7 within a FSR of MRR	>0.5 nm, incoherent	Uneven	○ limited density, lower SNR, high filter loss
Soliton microcomb (this work)	>20 lines 100 GHz FSR of the comb	<0.1 nm, mutually locked	>30 comb line having a power greater than -30 dBm, with integrated DFB laser	√ scalable, compact, on-chip, high OSNR

CW sources only provide one tunable wavelength at a time, to perform parallel MVM, this would require either one laser per MRR, which is clearly unscalable, or time-multiplexed access, which severely limits throughput and introduces timing complexity. Therefore, CW lasers are unsuitable for large-scale MRR computing.

While pulsed sources can produce broad spectra, they require WDM filters to extract individual wavelengths. These filters typically have bandwidths $\geq 0.3\text{--}0.5$ nm, which means fewer usable wavelength channels in a fixed band and larger guard bands between adjacent MRRs to avoid crosstalk. By contrast, our soliton microcomb provides linewidths < 0.1 nm, allowing denser packing of MRRs and scalable, low-crosstalk computing.

In our experiments, we utilized an integrated turnkey mode-locked microcomb that directly couples a commercial DFB laser to a Si_3N_4 microring resonator chip. The mode-locked microcomb is packaged in a standard butterfly package to ensure a fixed feedback phase and achieve reliable operation. This butterfly package can

be further integrated into a module with dimensions of $10\text{ cm} \times 7\text{ cm} \times 4\text{ cm}$, which includes a complete driving current source and temperature control unit, and is mounted on an electronic circuit board (Fig. S12a). The fully packaged optical frequency comb module is approximately the size of a computer mouse, making it highly portable (Fig. S12b). We tested the power deviation of eight channels of the optical frequency comb over a period of 60 minutes and performed statistical analysis. All comb channels maintained ultra-low power fluctuations below 0.1 dB, ensuring computational consistency and the capability for long-duration computations (Fig. S12b). Additionally, we can add an EDFA after the laser to increase the pump power, achieving greater soliton detuning to generate a broader spectrum. Fig. S12c shows a single soliton optical frequency comb with a spectral range exceeding 300 nm, where the signal-to-noise ratio meets the computational requirements for a spectral range over 150 nm. With a repetition frequency of 100 GHz, it covers more than 150 comb teeth, and its frequency range far exceeds the five FSRs of the MRR computational unit (Fig. S12d), enabling a high degree of parallelism. In the experiment used to perform the computational tasks, we employed the fully-integrated turnkey microcomb (Fig. 2b), whose spectral range spans 100 nm and covers two FSRs of the MRR computational unit. Therefore, in the experiment for sentiment analysis using PGRNN, we tested with two FSRs and completed the overall experiment through a cyclic process.

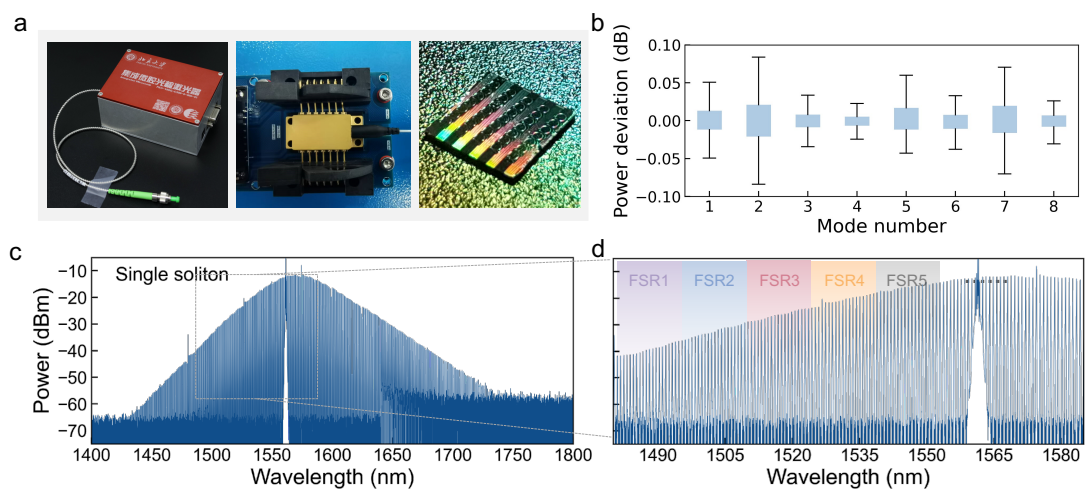


Fig. S12 | Integrated turnkey soliton microcomb. a, From left to right are a fully

packaged optical frequency comb module, the enclosed circuit board inside the module (including a complete driving current source and temperature control unit), and a Si_3N_4 chip in butterfly packaging mounted on the circuit board. **b**, The power deviation of eight channels of the microcomb over a period of 60 minutes. **c**, The microcomb generated by amplifying pump light through EDFA. **d**, Zoomed-in view of the spectrum close the pump wavelength of the microcomb.

For reliable MRR-based computation our integrated soliton microcomb must satisfy three quantitative criteria. 1) Comb Line Power: In our architecture, each comb line must pass through a sequence of optical components before reaching the chip—such as EDFAs, VOAs, WDMs, and grating couplers—all of which introduce insertion loss. If the initial power of a comb line is too low, the received signal at the chip will fall below the detection or modulation threshold, resulting in poor or unusable weight control in MRRs. Each comb tooth should emerge from the source at ≥ -30 dBm so that, after propagation through EDFAs, VOAs, WDMs and grating couplers, the optical field reaching the chip still exceeds the MRR modulation threshold; below this level we observe noisy, discontinuous voltage–transmission curves and weight instability. 2) Signal-to-Noise Ratio (SNR): The SNR of each comb line affects how reliably its intensity can represent an input signal amplitude. In our system, the intensity of each wavelength encodes input values in analog form, and is modulated by the MRR transmission to implement the weight. If the comb source has low SNR, the noise will be amplified by EDFAs, causing inconsistent minimum levels in MRR modulation. 3) Linewidth: It determines the channel spacing and total number of usable wavelengths in the system. In our architecture, MRRs are densely cascaded, and each ring must respond to a unique frequency. If spectral lines are too wide, they will overlap or require excessive spacing, limiting the number of addressable channels and increasing the device footprint. Narrow linewidths reduce the required channel spacing, increasing the number of MRRs that can be addressed in a given band. In essence, reliable photonic inference hinges on a comb source that offers sufficiently high line power, low noise,

and narrow linewidth to guarantee stable weight tuning and dense, interference-free wavelength multiplexing.

Supplementary Note 7 | Calibration of input encoding and weight deployment.

In the experiment, to ensure accurate input encoding and precise weight deployment on the MRRs, preliminary calibration of the optical frequency comb and the MRRs is required. Since different comb teeth of the optical frequency comb exhibit varying initial power levels, and the VOA does not modulate all wavelengths absolutely uniformly, we first perform a fixed-range scanning (e.g., 0 dB–40 dB) for each comb tooth required for computation, here we choose 8 comb teeth as samples to show the real calibration data during the experiment (Fig. S13). The power response of each comb is recorded as a function of attenuation. We read the minimum power value and maximum power value of each sample, select the maximum of all minimum values as the standard minimum value, and select the minimum of all maximum values as the standard maximum value. The standard minimum and maximum values are used as 0 and 1 of the input data code. Then we cut the attenuation power curve according to the standard minimum and maximum values, retain the data between the maximum and minimum values of power, and normalize its ordinate to between 0-1. Finally, we get the attenuation value-input encoding formula of each comb by 4-order polynomial fitting (Fig. S14). After fitting, we can correspond to the attenuation value of each comb teeth according to the required input coding value. This calibration method can completely avoid the error influence caused by the different initial power of optical frequency comb and the slight difference of VOA response to different wavelengths.

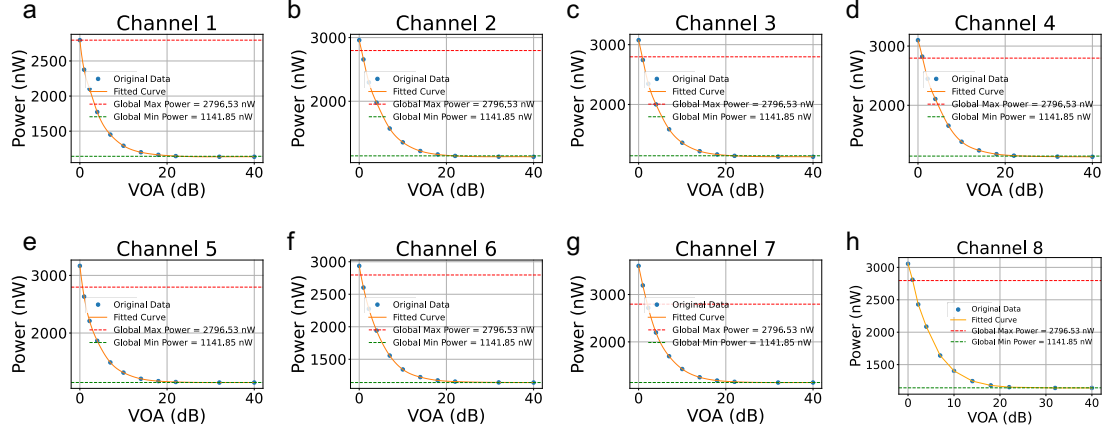


Fig. S13 | The power variation curve of the 8-channel comb teeth under modulation. The blue dots represent the original sample points, the orange line represents the fitting curve, the red dashed line marks the global maximum value, and the green dashed line marks the global minimum value.

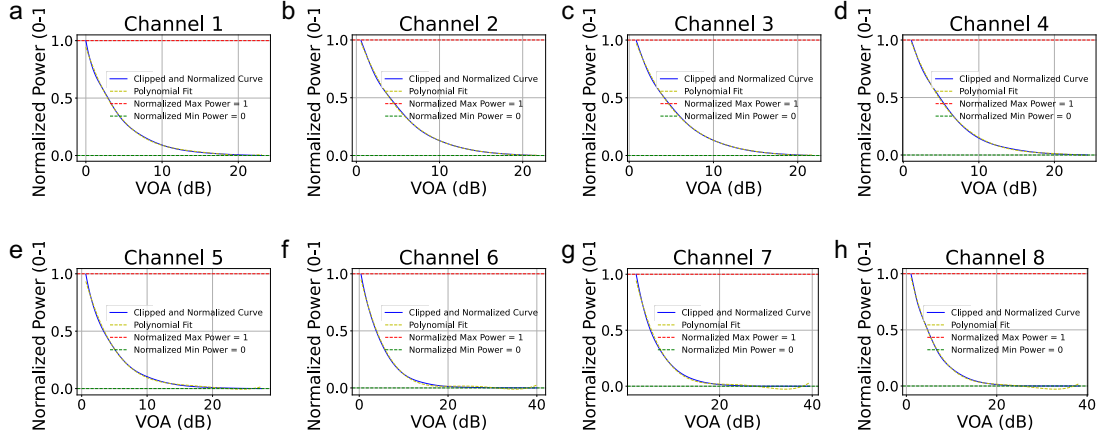


Fig. S14 | The normalized calibration power variation curve of the 8-channel comb teeth under modulation.

The maximum and minimum power of all channels are limited to the global maximum and minimum values, respectively. The blue curves represent the cropped and normalized power curves, the yellow dashed line represents the polynomial fitting curve, and the red and blue dashed lines mark the normalized minimum value of 0 and the maximum value of 1, respectively.

After calibrating the input, we proceed with fine-tuning the MRR weights using a

sequential sampling method. According to the fabrication specifications provided by our foundry, the precision for the coupling gap is $\pm 5\%$. This variation directly affects the coupling strength between the waveguide and the MRR, which in turn determines the resonance depth of each MRR. Consequently, even when two MRRs are tuned to align with a comb line (i.e., theoretically reach their transmission minimum), their transmission depths can differ, leading to inaccuracies in the realized weights. In addition to coupling gaps, the geometry and positioning of the metal electrodes used for thermal tuning also introduce variation. For instance, electrode coverage and thermal contact can differ slightly across devices, resulting in different voltage-resonance tuning efficiencies. Therefore, before computation, every MRR requires calibration. We first determine the approximate voltage range needed for each MRR to achieve resonance at its assigned input wavelength. Here, eight rings are selected as samples (Fig. S15). Their initial voltages are (5.2 V, 5.2 V, 5.3 V, 4.3 V, 4.7 V, 5.1 V, 3.0 V, 4.1 V). These initial voltages correspond to the eight MRRs in a row at the wavelengths corresponding to the eight adjacent comb teeth. It can be seen that their voltages do not increase regularly with the increase of resonant frequency, indicating that each MRR has a unique response to the voltage. Therefore, it is necessary to calibrate each microcavity separately. As shown in Figure S15b, (i) at $V = 4.1$ V, the MRR resonance lies to the right of the comb line (yielding a logic output of 1); (ii) at $V = 4.58$ V, the resonance is aligned with the comb line (logic output 0); and (iii) at $V = 4.9$ V, the resonance moves to the left of the comb line (logic output returns to 1). ④ A continuous scan from 4.1 V to 4.58 V in 0.01 V steps was performed to generate the voltage-output response curve shown in Figure S15c. Then we automatically perform a fine voltage scan from the initial voltage with a scan step of 0.01 V, and let the system stop scanning the MRR after detecting the inflection point of the minimum value to obtain a resonance curve similar to the half Lorentz shape (Fig. S16). Since manufacturing defects not only affect the resonant frequency of each MRR, but also affect its quality factor (Q factor), we apply the global normalization strategy to unify the weight scaling of all MRRs. Specifically, according to the automatically recorded

maximum and minimum transmission voltage values, the system takes the minimum and maximum values in all MRRs as the common maximum transmission values, and the maximum and minimum values as the common minimum transmission values (Fig. S17). Once these global reference values are determined, they are mapped to the voltage level, and a function fitting model is used to establish a lookup table for converting the required weight values into voltage settings. This automatic calibration of pipeline voltage scanning, extreme value detection, function fitting and weight look-up table generation can achieve seamless and efficient weight calibration without excessive manual intervention.

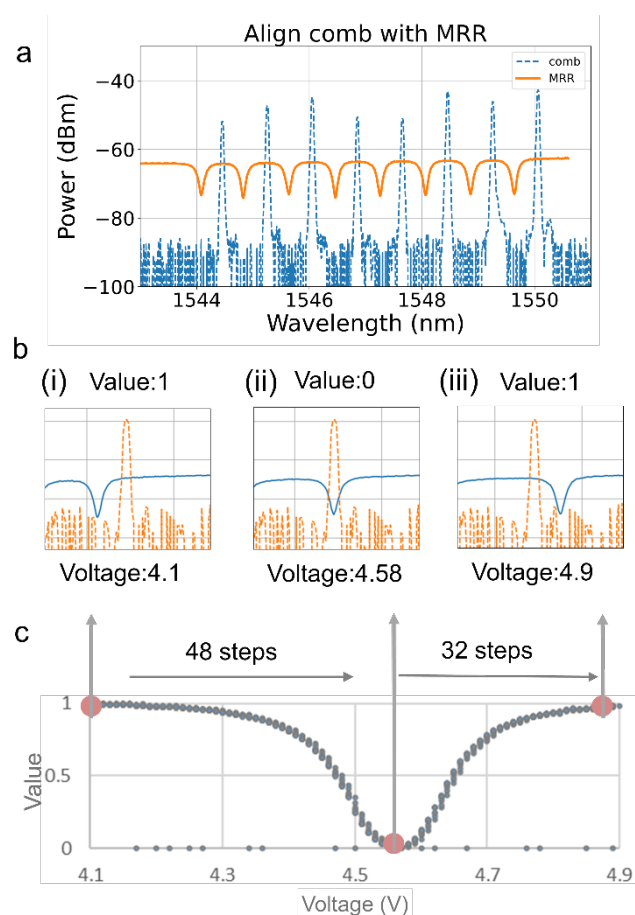


Fig. S15 | Demonstration of the fine-tuned voltage control process. (a) eight MRRs voltage-biased to align their resonant wavelengths with the wavelengths of the eight comb teeth. (b) a single MRR channel correspond to three specific voltage points. (c) voltage-output response curve of one MRR.

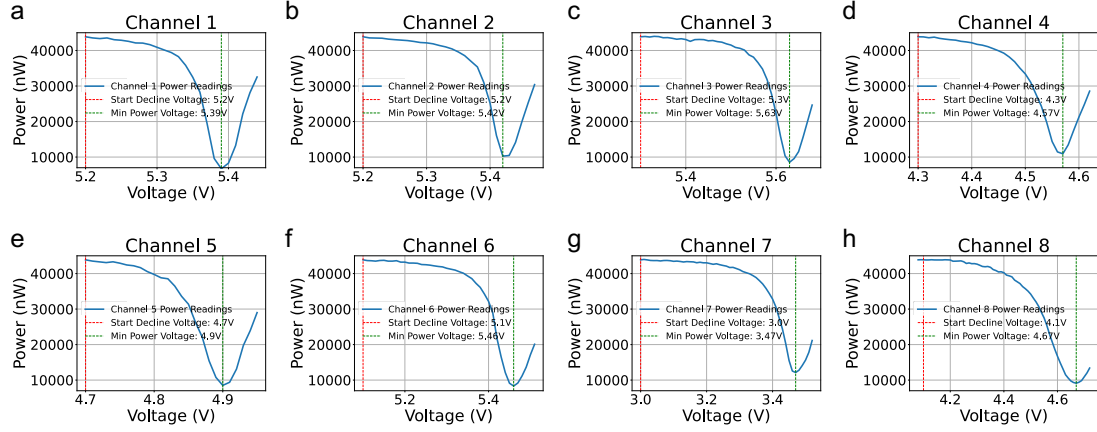


Fig. S16 | Output power curve of MRR modulated by voltage (before unified). The power variation with voltage for 8 MRRs during calibration is recorded. The blue curve represents the values read by the power meter, the red dashed line indicates the voltage at which the power starts to decrease as automatically detected by the system, and the green dashed line shows the voltage corresponding to the point of minimum power as automatically detected by the system.

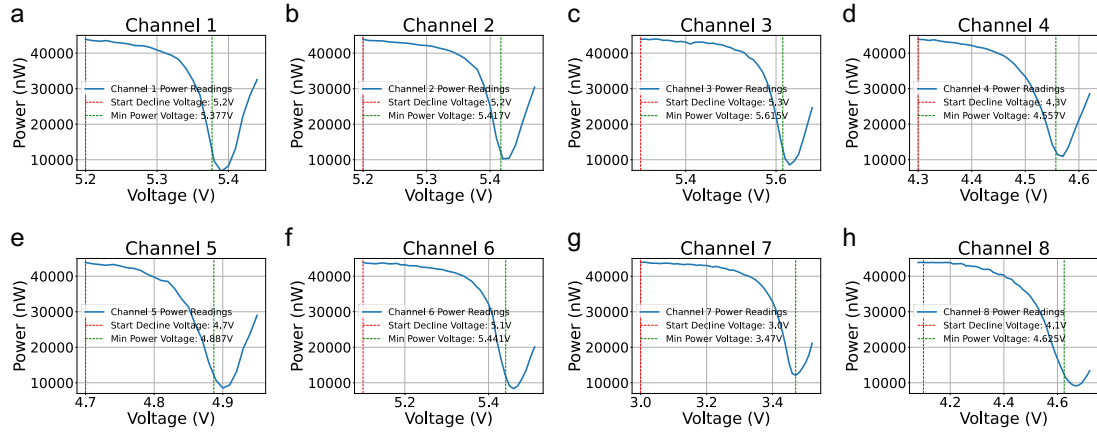


Fig. S17 | Output power curve of MRR modulated by voltage (after unified). The green dashed line represents the control voltage corresponding to the re-labeled minimum value after selecting a unified minimum. The rest of the content is the same as in the Fig. S16.

Furthermore, we also tested one MRR response curve under voltage regulation under different comb power (0-40dB). After normalization, they are also completely consistent, which indicates that the above fitted voltage transmittance formula is still

used under different input power (Fig. S18).

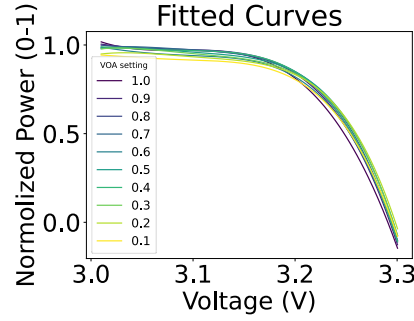


Fig. S18 | Normalized power output of the MRR under different inputs. The curves in different colors represent input power levels ranging from 100% to 10%. The curves maintain a consistent shape across all input levels. The coefficients obtained from the quartic polynomial fit (from highest to lowest order) are: -27.57, 229.64, -550.15, 75.72, and 758.10.

Following the calibration of the MRR, the next step is to calibrate the MZI. In this process, we have chosen four MZI samples and measured their response curves under voltage control ranging from 0 to 8 V, with a step size of 0.05 V (Fig. S19). Analogous to the calibration method employed for the MRR, we analyze each MZI curve to determine its maximum and minimum value ranges. Subsequently, we fit a voltage-transmittance function based on these data points to construct the MZI lookup table.

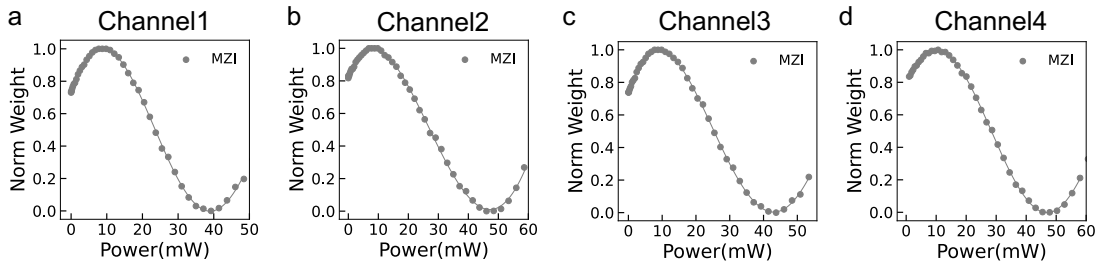


Fig. S19 | Normalized weight of the MZI under modulation. Four MZI channels were selected as samples, and the section within the 0-8 V modulation range that includes the first approximately linear region was extracted. The overall linear trend of the four channels is consistent, however, their x-axis values, i.e., the modulation voltage corresponding to the normalized weight range of 0-1, differ. This behavior is similar to that of the MRR, where the differences in modulation for each unit are caused by

fabrication errors.

Supplementary Note 8 | Supplementary training process and sample results for speech recognition tasks.

Sentiment analysis is an important branch of NLP, and the process by which this network addresses sentiment analysis also lays the groundwork for tackling more complex NLP tasks, such as translation and text continuation.

① Data Filtering and Preprocessing:

To enhance the efficiency and accuracy of model training, we conducted data filtering and preprocessing:

- Data Filtering: Considering the scale of the network and training duration, we set a threshold for the number of words to filter out texts exceeding the maximum word count (set to 60 in the program).

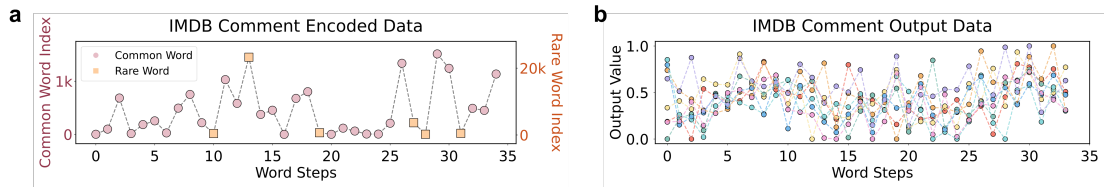


Fig. S20 | Vocabulary Construction. (a) Word encoding for an input sentence corresponding to a sample in the IMDB dataset. (b) The 8 feature vector outputs at each word step after the sample passes through the PGRNN.

- Vocabulary Construction: The first step involves mapping words in the text to unique index values to construct a vocabulary (partial vocabulary example: the: 2, and: 3, a: 4, of: 5, to: 6, is: 7, in: 8, i: 9, this: 10, that: 11), as shown in the Fig. S20a. The smaller the index, the higher the frequency of the word's occurrence. Indexing starts from 2, reserving 0 for padding tokens and 1 for unknown words (<unk>). Thus, the original text "This movie was excellent! The performances were oscar-worthy." is processed into the index sequence "[20, 50, 40, 80, 2, 55, 66, 109, 120]" (Fig. S17b), with the label "1" indicating a positive sentiment for the review.
- Data Processing: After converting the text data into PyTorch Dataset objects, we

first pad the sequences to ensure consistent length within each batch, then normalize the data to correspond to the optical input. Finally, an embedding layer maps each word into an 8-featured vector.

② Model Architecture and evaluation:

Based on the previously introduced Pseudo-Gated Recurrent Neural Network (PGRNN), we customized the model's forward propagation network, with trainable parameters including the weights of MRRs and MZIs in the photonic network. Since the ultimate target labels are 0 or 1, we chose the binary cross-entropy loss function, given by the formula:

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)], \quad (9)$$

where N is the number of samples, y_i is the true label of the i -th sample, and p_i is the probability predicted by the model that the i -th sample is labeled as 1. The loss value calculated by the loss function guides the Adam optimizer in updating the weights. The learning rate is set to 0.0001, the batch size is 64, and after 800 training epochs, the network converges, achieving a final training accuracy of 99.23% and a testing accuracy of 82.35%. Table. S2 presents a selection of samples for further illustration of the training results.

Table. S2 | Sample cases from the IMDB dataset. Representative samples were selected, including those with rare words, sarcastic expressions, neutral attitudes, etc., along with their true labels and predicted labels. The characteristics of each sample are detailed in the table.

Comment1	True Label	Predict Value	Predict Label
a gem of a cartoon from the silent <UNK> was re-discovered by cartoon network, and was broadcast for likely the first time in decades, if ever. what makes this so enjoyable are the varied <UNK> fairbanks is attacked by giant <UNK> will <UNK> pays a visit as <UNK> of static <UNK> as well as appearances by chaplin, keaton,	1	0.53	1
Sample Characteristics			
The use of rare words led to many words being recorded as <UNK>, and the final predicted value slightly exceeded the threshold to yield the correct result.			

and william s. hart. the image of chewing gum <UNK> the shoes of the populace (a <UNK> idea for <UNK> <UNK> <UNK> boss) cannot be <UNK> must be viewed. a terrific cultural gem.			
Comment2	True Label	Predict Value	Predict Label
errol flynn at his best as robin hood of the west, fighting military red tape, <UNK> , indians and <UNK> business crooks <UNK> to his great and final heroic end. not to forget the ever reliable o. de <UNK> as lady mary of the west. never try to link this story to the facts and the real persons, it doesn't work out. just enjoy it, because nobody ever claimed to make documentaries when raoul walsh and errol flynn <UNK>.	1	0.20	0
Sample Characteristics			
The emotional tone is not clearly expressed, and many neutral, declarative statements can easily lead to misunderstandings, ultimately resulting in incorrect judgment.			
Comment3	True Label	Predict Value	Predict Label
This isn't a bad movie. It's fun to watch for the first time. However it has absolutely no<UNK>value at all. When you try to watch it again it gets so boring you have to turn it off. I give this movie a 6 out of 10.	0	0.45	0
Sample Characteristics			
This comment is difficult to analyze. The review states that the movie is not worth watching a second time, but it was fun the first time, and the rating is above 5. However, the dataset classifies it as a negative review, and our model also predicted it as negative.			
Comment4	True Label	Predict Value	Predict Label
If you like bad movies (and you must to watch this one) here's a good one. Not quite as funny as the first, but much lower quality. A <UNK> for fans of Jack Frost as well as anyone up for a good laugh at the writing.	0	0.33	0
Sample Characteristics			
The review uses a sarcastic tone, which can easily lead to misinterpretation by machine learning models, but we predicted the correct outcome.			

Supplementary Note 9 | FCNN-CNN-PGRNN network schematic and detailed completion process of speech recognition tasks.

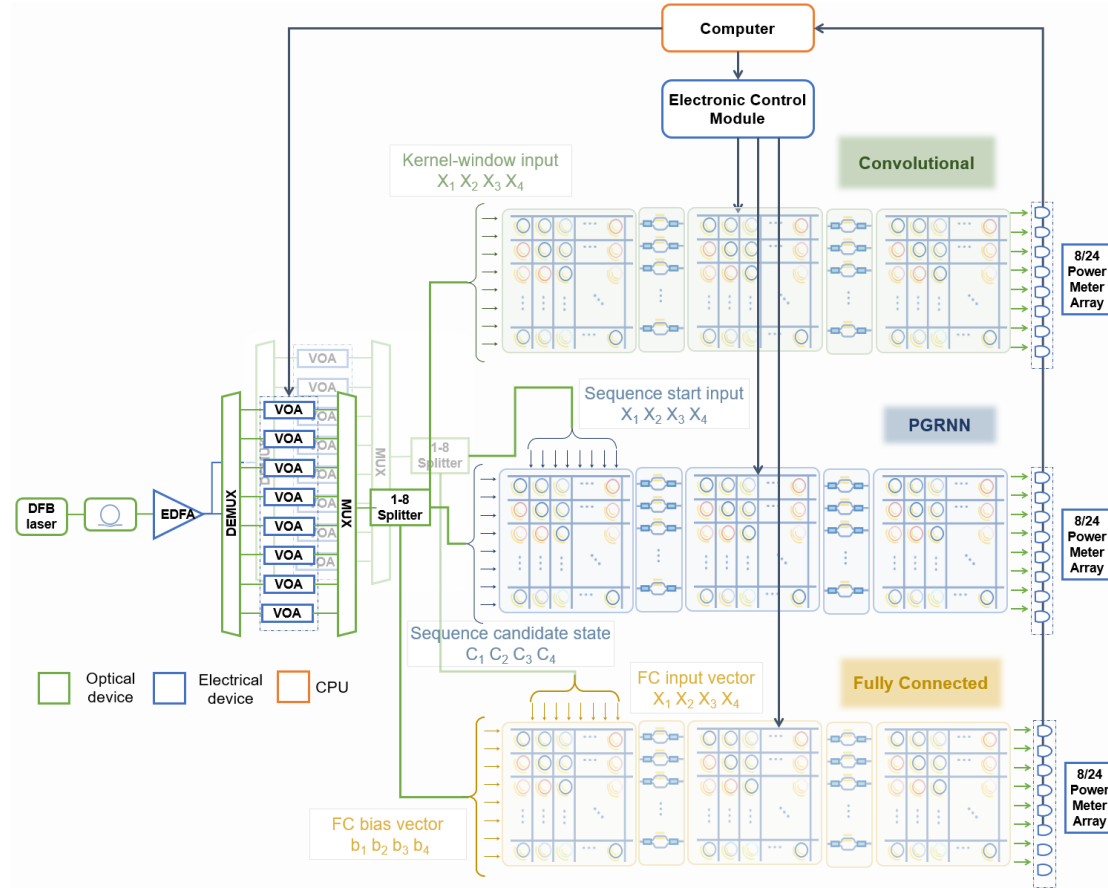


Fig. S21 | Physical architecture of the photonic chip used for speech recognition.

As illustrated in Fig. S21, the photonic physical architecture includes three part of arrays that are individually configured to implement CNN, PGRNN, and FCNN layers, respectively. The computation proceeds in sequential stages: first, the input features are processed by the CNN module (top layer in the schematic), whose outputs are optically detected and routed back to the CPU. There, a VOA-encoding unit adjusts the input signal and directs it to the PGRNN array. This process is repeated after PGRNN inference to prepare inputs for the final FCNN classification stage. This flow of compute-detect-redistribute-compute ensures that each sub-network receives appropriately formatted input and enables multiple model types to be executed in a coordinated manner.

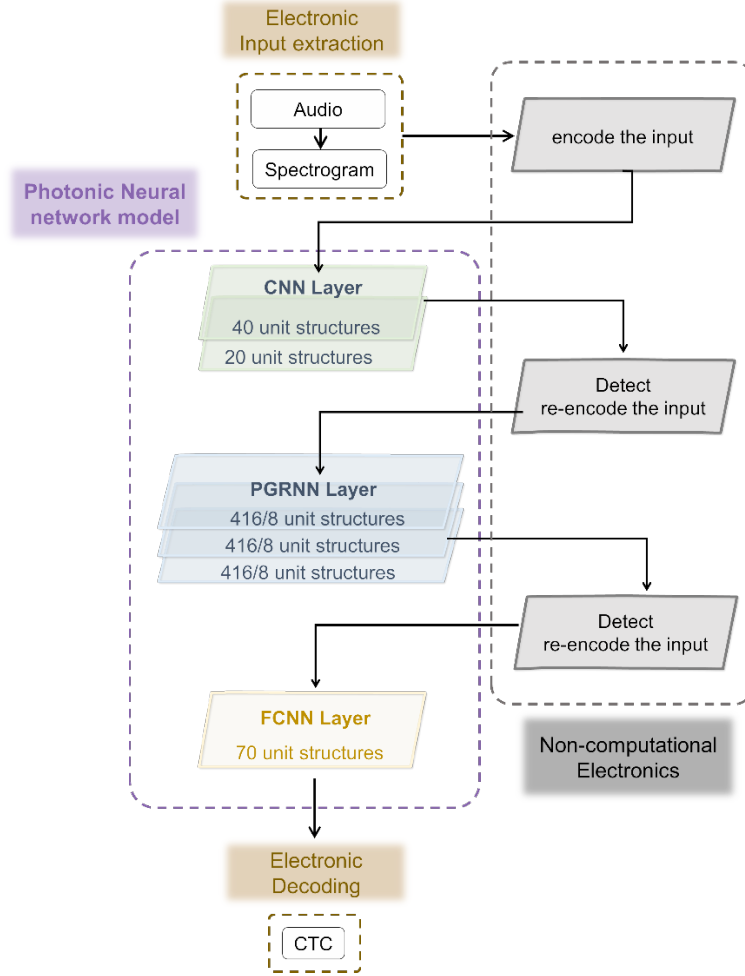


Fig. S22 | Layer-wise execution flow and data movement throughout the system.

Figure S22 present the layer-wise execution flow and data movement throughout the system, highlighting which steps are performed by the photonic chip and which are handled by the electronic components.

① Photonic Chip:

As shown in the Fig. S22, the core neural computation is carried out optically. CNN layers extract local time-frequency features from the input vector sequence. The PGRNN modules handle temporal inference across frames, capturing sequential dependencies. Finally, the FCNN performs mapping to a 32-class output space (representing character probabilities). These three components form the bulk of the compute workload and are handled entirely by the photonic neural network.

② Electronic Components:

- Input Extraction (Fig. S22, brown blocks): Audio signals are pre-processed on the CPU using standard signal processing techniques (e.g., Fast Fourier Transform) to convert raw audio into feature vectors. This is a necessary step because current photonic chips do not directly accept analog acoustic signals. However, in future work, this stage could be replaced with photonic or optoelectronic frontend components capable of direct sound acquisition. Additionally, the final CTC (Connectionist Temporal Classification) decoding step is performed electronically. This decoding does not involve neural computation; it simply translates the predicted output probabilities (from the result calculated by the photonic FCNN module) into character sequences based on sequence alignment rules. Its role is evaluative, not computational.
- Readout and Redistribution (Fig.S22, gray blocks): Since different models executed on the same shared photonic hardware require different input encoding scheme, the output of one photonic module cannot be directly passed to the next. Instead, the CPU assists by reformatting and re-encoding the detected signals to match the configuration required for the subsequent sub-network. This enables routing the outputs of one model to the inputs of the next, potentially with format or timing adjustments. Importantly, no learning or inference is done in the CPU during this stage, the electronics serve only as signal carriers and modulators.

In summary, While the system is not fully all-optical, the photonic chip handles the most computation-intensive parts of the pipeline. Electronic components are limited to peripheral signal conditioning, re-encoding, and input/output interfacing. We believe this architecture demonstrates a practical and scalable hybrid strategy where photonics is used efficiently for core AI computations.

The network employs Short-Time Fourier Transform (STFT) for preprocessing audio data and utilizes a PGRNN for feature extraction and sequence modeling. The model is optimized using the Connectionist Temporal Classification (CTC) loss

function, ultimately achieving end-to-end speech recognition. We utilized the LJ-Speech dataset, a publicly available speech dataset comprising 13,100 short audio clips, each accompanied by corresponding text transcriptions. The dataset was downloaded and loaded using TensorFlow's `keras.utils.get_file` function to ensure data integrity and accessibility. The completion of the speech recognition task can be divided into the following components.

① Data Filtering and Preprocessing:

To enhance the efficiency and accuracy of model training, we conducted data filtering and preprocessing:

- **Short Audio Filtering:** By calculating the duration of each audio clip, we filtered out clips with durations exceeding the predefined maximum value (20 seconds) to avoid overburdening the model training process.
- **Dataset Splitting:** The dataset was divided into training, validation, and test sets, which were used for model training, hyperparameter tuning, and final performance evaluation, respectively.
- **Data Preprocessing:** Initially, we performed time-frequency analysis on the input data by transforming each audio signal using STFT. While traditional Fourier Transform can analyze the frequency components of a signal, it fails to capture the temporal variations of these frequencies. STFT overcomes this limitation by segmenting the signal into multiple short time windows and applying Fourier Transform to each segment, thereby providing both temporal and spectral information. This time-frequency analysis method is particularly suitable for processing speech signals, as it captures the dynamic characteristics of speech, such as pitch and tempo variations. The STFT converts the audio signal into a spectrogram, a two-dimensional representation where the horizontal axis denotes time, the vertical axis represents frequency, and the color or grayscale indicates the amplitude of the frequencies (Fig. S23). In the STFT operation, “`frame_length`” denotes the length of each STFT window in terms of sample points (set to 256 in the code), “`frame_step`” represents the stride between adjacent windows (set to 160

in the code), and “fft_length” indicates the length of the Fast Fourier Transform (FFT) (set to 384 in the code). The FFT length determines the frequency resolution of the spectrogram. Prior to STFT, the audio signal is one-dimensional, with its dimension determined by the number of sample points. After STFT, the audio signal is transformed into a two-dimensional spectrogram with dimensions of (time steps, frequency bins). The number of time steps is calculated as $(\text{sample points} - \text{frame_length}) / (\text{frame_step}) + 1$, and the number of frequency bins is $\text{fft_length} / 2 + 1 = 193$. Therefore, the sample dimension after STFT becomes (time steps, 193). Finally, the complex spectrogram is converted to a magnitude spectrogram by taking the modulus and then normalized for use as the input signal.

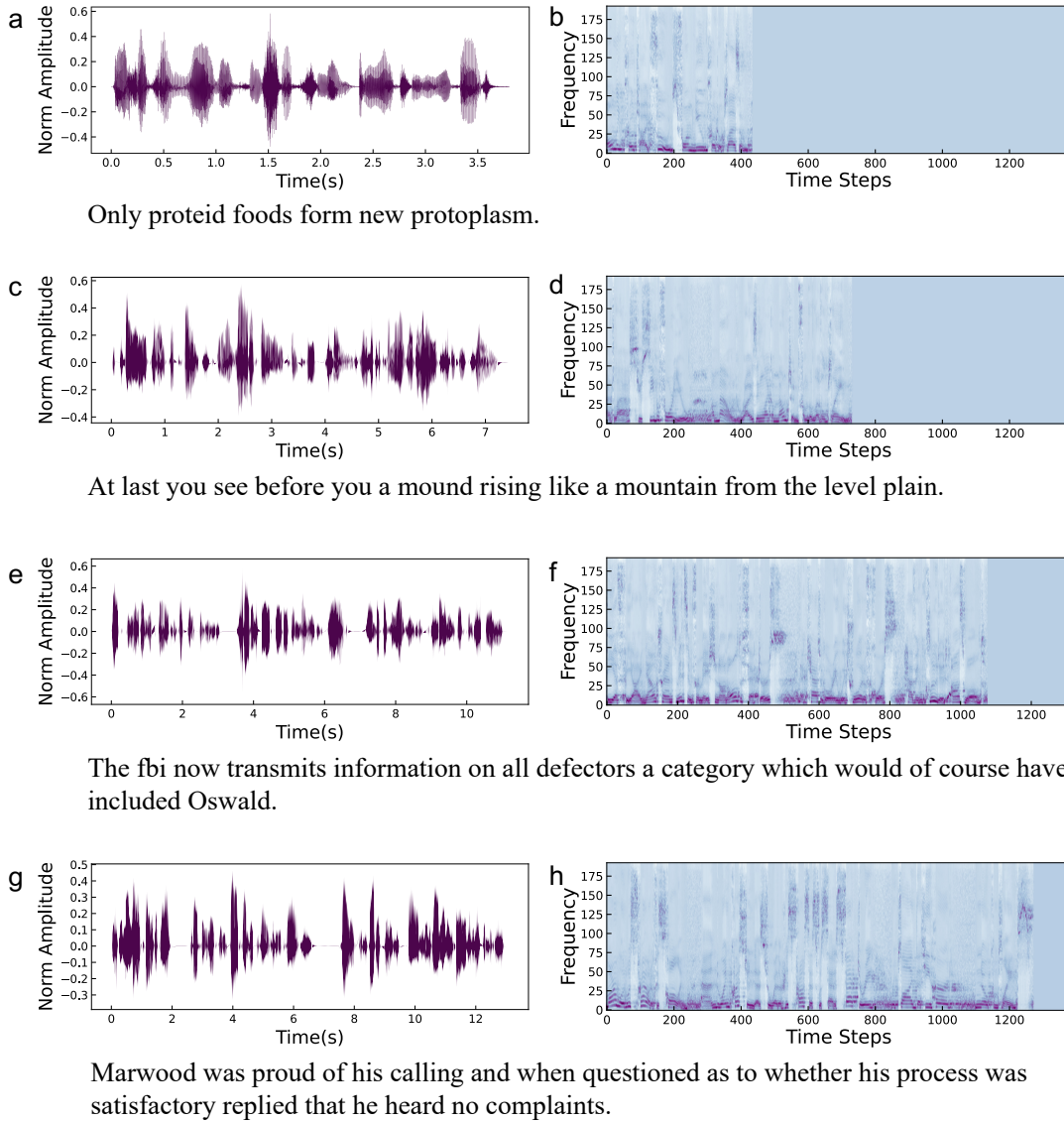


Fig. S23 | Sample cases from the LJ-speech dataset. Four sets of original audio

samples (a, c, e, g), along with their corresponding text and spectrogram after STFT (b, d, f, h).

② Model Architecture:

- Vocabulary Construction: We defined a vocabulary comprising English letters, punctuation marks, and spaces ("abcdefghijklmnopqrstuvwxyz?!/"), which was mapped to numerical representations using the “StringLookup” layer.
- Network Structure: The first convolutional layer has 32 filters with a kernel size of (8, 30) and a stride of (2, 4). The second convolutional layer also has 32 filters with a kernel size of (8, 15) and a stride of (1, 4). After these two convolutional layers, the sample dimension is transformed from (time steps, 193) to (time steps, 13, 32). The output is then reshaped to (time steps, 13×32), i.e., (time steps, 416) and fed into a three-layer PGRNN with a hidden layer dimension of 416. The output from the PGRNN is passed through a FCNN to reduce the dimension from 416 to 32 for the final decision output. Figure S25 shows the complete process from the input raw signal to the RNN processing result to the FC processing result.
- Training with CTC Loss: We employed the CTC loss function to train the model. CTC is particularly suited for sequence alignment problems, as it does not require strict matching or alignment of input and output sequence lengths. The output layer consists of a Softmax layer with a size of $n \text{ classes} + 1$ (where n classes are the size of the character set, and the additional category represents the blank symbol) to generate the probability distribution at each time step. The blank symbol allows the model to produce no character output at certain time steps, addressing the alignment issue between characters.

③ Model Training and Evaluation:

Model training was conducted using batch gradient descent, iteratively optimizing the model weights through the training dataset. At the end of each training epoch, we calculated the Word Error Rate (WER) on the training and validation sets using a custom callback function and recorded the loss values and WER changes throughout

the training process. After 200 epochs, the loss and WER curves for the training and validation sets are shown in Fig. S24, with the test WER reaching 0.55.

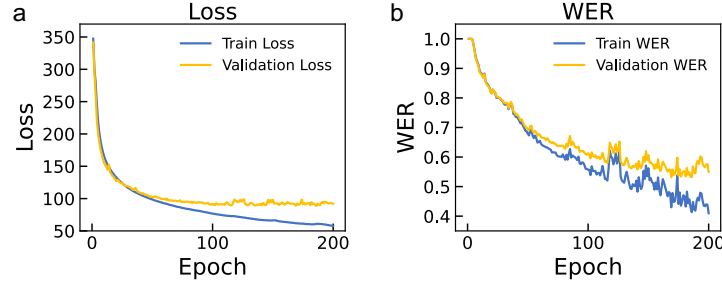


Fig. S24 | Loss and WER on the LJ-Speech dataset. a, The loss curves of the training set and validation set as a function of the number of training epochs. **b,** The WER curves of the training set and validation set as a function of the number of training epochs.

The lower the WER, the higher the recognition accuracy of the model. Based on the above task-oriented training workflow, we illustrate the process of a single sample from raw audio input to final predicted text output (Fig. S25). In addition, we present more test samples along with their ground-truth labels, providing an intuitive reflection of the model's performance (Fig. S26).

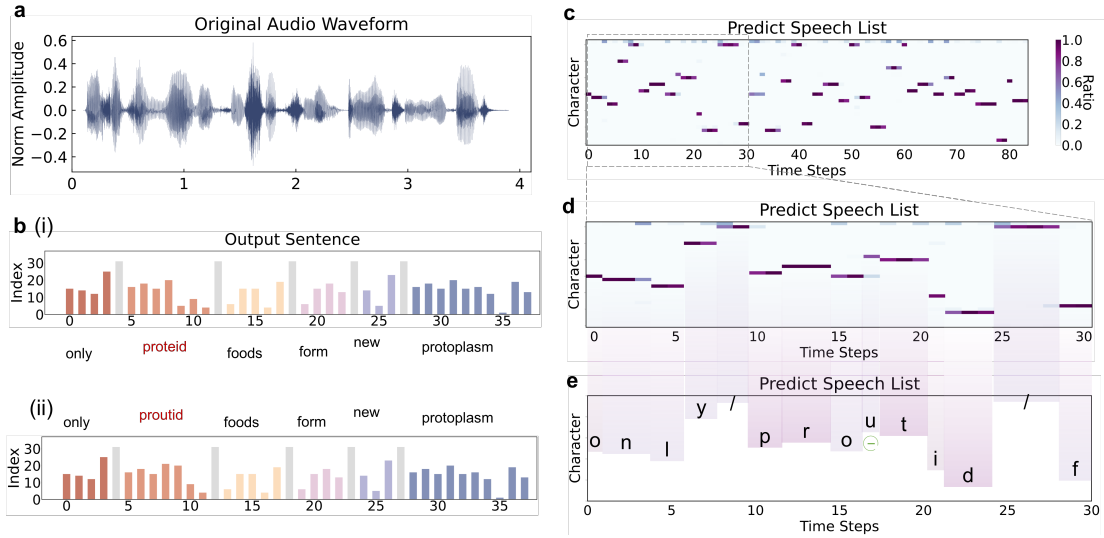


Fig. S25 | Speech recognition tasks. (a) The input encoding process and neural network inference flow for the speech recognition task. **(b)** Raw audio waveforms for two samples from the LJ Speech dataset. **(c)** The corresponding true texts (i) and predicted texts (ii). The error in (ii) is one substitution (replacing the word "proteid"

with "proutid", marked in red in (i) and (ii)). (d) The character probabilities at each time step for the two samples, with the y-axis representing the 26 English letters and common punctuation marks, totaling 32 characters. (e) Zoomed-in view of the character probabilities for the first 30-time steps of the two samples and their corresponding final predicted output characters (H-I).

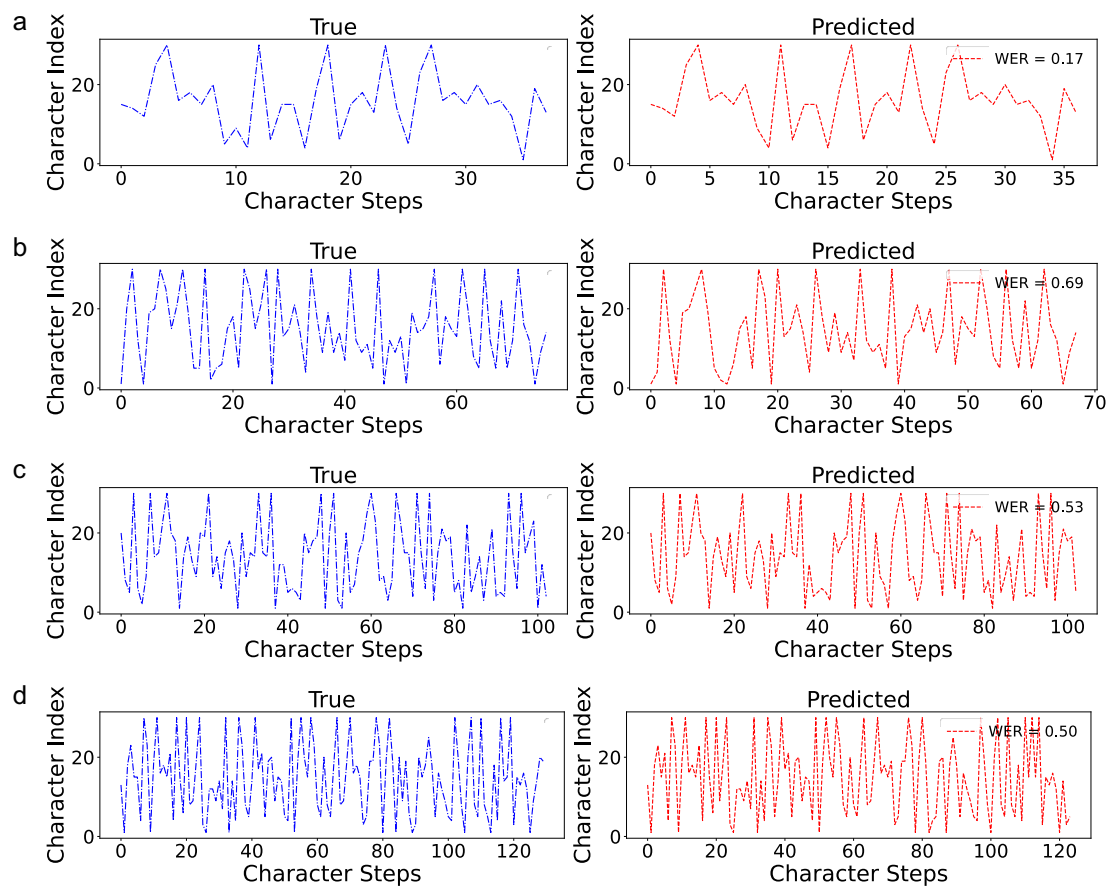


Fig. S26 | Comparison between the target and predicted texts. (a-d) corresponds to the true outputs and predicted outputs of the four samples in Fig. S23, with WER values of 0.17, 0.69, 0.53, and 0.50, respectively.

Supplementary Note 10 | Area/Power Efficiency and Computing Accuracy of the Photonic Computing Chip.

To evaluate the performance of the proposed chip architecture, we calculate its computing speed and area efficiency. Considering an $n \times n$ MRR crossbar array, the computing speed, expressed in tera-operations per second (TOPS), can be quantitatively determined using the following formula:

$$\text{TOPS} = 2N^2 f_{clk} \times 10^{-12}, \quad (10)$$

where N denotes the dimension of the MRR crossbar array, and f_{clk} represents the clock frequency of the modulator. The factor of 2 accounts for the combined operations of multiplication and addition, which are essential arithmetic operations in most computational tasks. In our design, since both ports of the cross-coupled waveguides can carry information, the computing speed is effectively doubled to $4N^2 f_{clk}$. Since the Q-factor is the ratio of the confined optical intensity to the decayed optical intensity in the resonator, we can define the photon lifetime t as

$$t = \frac{\lambda_{res} Q}{2\pi c}, \quad (11)$$

where λ_{res} is the resonance wavelength of the MRR, and c is the speed of light. t is associated with the bandwidth of the MRR, the reciprocal of t reaches the THz scale, significantly surpassing the modulation speeds of existing modulators. Consequently, the computing speed of the chip is predominantly governed by the clock frequency of the modulator. We configured an 8×8 MRR array and an array of 8 Mach-Zehnder Interferometers (MZIs) to form a single-layer neural network. The total number of operations is calculated as:

$$\text{Operations} = 4 \times 8 \times 8 + 1 \times 8 = 256, \quad (12)$$

thus, the computing speed is expressed as $256 f_{clk} \times 10^{-12}$.

In the current experiments, the computing speed is limited by the modulation speed of the VOA, which operates on the millisecond scale. However, by replacing the low-speed VOA with a high-speed LiNbO₃ electro-optic modulator, the computing speed can reach 2.56 TOPS at a typical clock frequency of 10 GHz. Utilizing an even higher-speed modulator operating at 100 GHz, the computing speed is projected to escalate to

25.6 TOPS. The structural dimensions of the single-layer neural network, comprising the MRR and MZI arrays, measure $1.12 \text{ mm} \times 0.96 \text{ mm}$. Given this footprint, the area efficiency can be calculated as:

$$\text{Area Efficiency} = \frac{\text{Computing Speed}}{\text{Area}} = \frac{2.56 \text{ TOPS}}{1.12 \text{ mm} \times 0.96 \text{ mm}} \approx 2.45 \text{ TOPS/mm}^2, (13)$$

at a typical clock frequency of 10 GHz.

The power consumption of our proposed computing system is mainly from six aspects: fully-integrated soliton microcomb, EDFA, data modulation, silicon photonic chip, modulator drivers, power meter and CPU.

Table. S3 | Energy consumption breakdown of the photonic computing chip.

	Components	Specifications	Power (W)
Integrated	Fully-integrated soliton microcomb	including the temperature controlling module and the circuit boards)	~5
	Silicon photonic chip	32 input/24 output	0
Non-integrated	VOA	32-channel	~4
	EDFA	ACC mode	~40
	Thermal phase modulator	192 MRR heaters and 16 MZI heaters	~15
	Power meter	24-channel	~15
CPU			~100
Total power consumption			~179

Given this power consumption, the energy efficiency can be calculated as:

$$\text{Energy Efficiency} = \frac{\text{Computing Speed}}{\text{Power Consumption}} = \frac{2.56 \text{ TOPS}}{179 \text{ W}} \approx 0.014 \text{ TOPS/W}. (14)$$

To further reduce the power consumption of the experimental system, future efforts should focus on integrating external desktop devices onto the chip. For example, this could be achieved through the use of circuit-based EDFA, on-chip

semiconductor optical amplifiers (SOAs) to fully integrate the light source with the computational chip. Additionally, with the development of hybrid and monolithic integration technologies, the light source, photonic computing chip, and external circuit modules can be further highly integrated to reduce power consumption.

The accuracy of the MRR weight bank is intrinsically linked to the dynamic range and the effective bit resolution of the system, which are determined by the measurable weight (w) and the target weight ($\hat{w}$). The number of discrete levels within this range, denoted as N , is calculated using the following formula²:

$$N = \frac{w_{max} - w_{min}}{2|w - \hat{w}|_{max}}, \quad (15)$$

where w_{max} and w_{min} represent the maximum and minimum achievable weights, respectively, while $|w - \hat{w}|_{max}$ signifies the maximum deviation from the commanded weights. This deviation can be either positive or negative, thus the minimum discernible weight interval is $2|w - \hat{w}|_{max}$. In our experimental case, we have $w_{max}=1$, $w_{min}=0$, and $|w - \hat{w}|_{max} \approx 0.01$. Substituting these values into the equation, we obtain:

$$N = \frac{1 - 0}{2 \times 0.01} = 50. \quad (16)$$

This indicates that there are 50 discrete levels within the dynamic range of the MRR weight bank. The effective bit resolution, which is a measure of the system's precision, can be derived from the number of discrete levels N using the logarithmic function:

$$\log_2(N) = \log_2\left(\frac{1 - 0}{2 \times 0.01}\right) \approx 5.64\text{bits}. \quad (17)$$

The precision of the MRR and the size of the MRR matrix exhibit a trade-off relationship. To construct a larger $N \times N$ MRR matrix, it is necessary to connect more MRRs in series within a single row. To ensure that each MRR can be independently controlled by different wavelengths, an MRR must be able to accommodate N resonance peaks within one FSR. This requirement necessitates a higher Q value to guarantee that there is no crosstalk between the N adjacent resonance peaks. However, a higher Q value can lead to a decrease in computational

precision. In our experimental design, the MRR is capable of accommodating up to 17 resonance peaks within one FSR and achieves a computational precision of 5.64 bits.

The accuracy reported above reflects the ideal conditions; however, in practical applications, various perturbations such as thermal drift, can affect the computational precision. As shown in Figure S27a, the spatial distribution of inference error across the output array, the measurement time in this experiment was approximately 4 minutes. Since measurements are scanned from top to bottom, the bottom rows are more affected by accumulated drift, and the error magnitude increases gradually. However, the overall error remained within acceptable limits, and recalibration was performed after each full input, thus bounding the total drift error per task. To further clarify, we include two supporting plots: As shown in Figure S27b: Drift-induced error as a function of measurement time, showing an increasing trend toward the latter half of the sequence. To further visualize the drift phenomenon, we plotted the variation of cumulative error over time, which is calculated as the accumulated absolute deviation of the output at each step. As shown in Figure S27c, where early-stage fluctuations are minor but a more rapid rise occurs later. Importantly, recalibration occurred before this trend led to significant degradation.

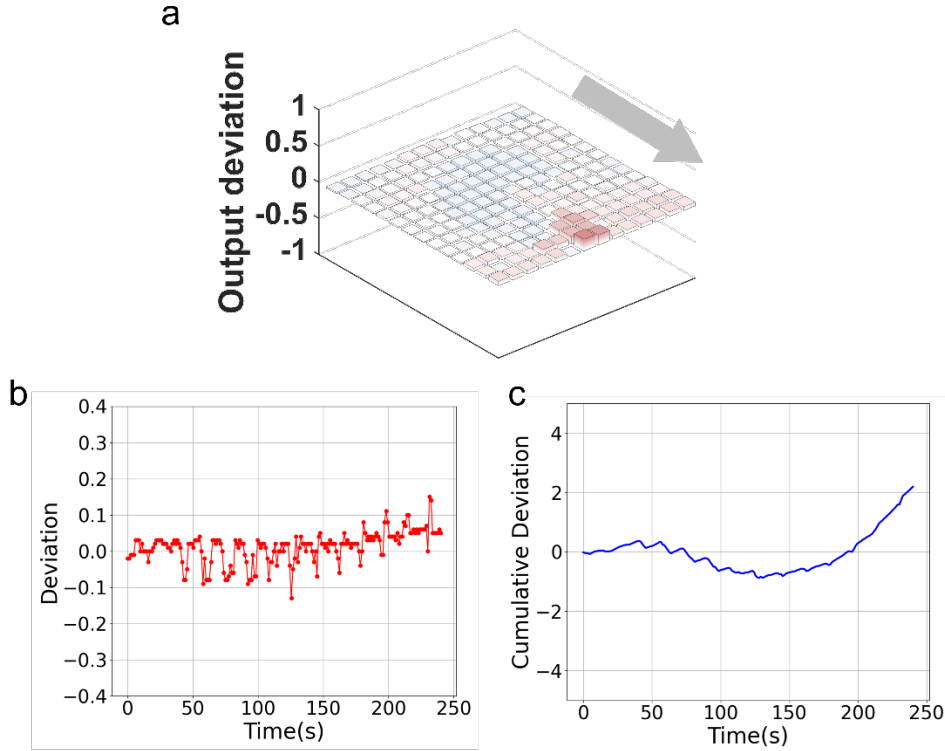


Fig. S27 | Impact of thermal drift on computation. (a) Deviation in output results across computation steps, following the direction indicated by the arrow. (b) Temporal evolution of output deviation for a test step. (c) Accumulated deviation over consecutive computation steps.

Besides thermal drift, we should clarify that whether thermo-optic modulation rate impact the task performance, especially for the dynamic tasks. Although thermo-optic tuning generally operates on the microsecond scale, in our experimental design, this modulation rate does not introduce latency issues for the dynamic tasks because the fixed weights during a specific inference task and relatively slow modulation of inputs.

But calculation speed determines the degree of temperature drift. As a result, completing a full inference task takes a relatively long time, exacerbating the impact of thermal drift. If we upgrade our hardware, e.g., by employing commercially available high-speed lithium niobate modulators (such as Thorlabs LNA2124) and AWGs operating at 10 GHz or higher, we could significantly accelerate the overall computation process. This would shorten the inference time per sample and reduce

the temporal window over which thermal drift can accumulate, thereby improving the effective inference accuracy. Besides, optimizing software algorithms, and implementing robust thermal control also can be concerned to improve the calculate accuracy.

Supplementary Note 11 | Scalability Potential of the Photonic Computing Chip.

The scalability potential of the photonic computing chip is primarily influenced by three key aspects: 1) optical components characteristic, including the capabilities of (i) the input optical source; (ii) the WDM device (iii) the performance of MRR, fundamentally determining the upper bound of chip performance; 2) thermo-optical tuning technology, crucial for reconfigurable photonic chips.

1) From the aspect of the optical components characteristic.

(i) The input optical source:

First, focusing exclusively on the performance of optical frequency combs, our current work utilizes a single soliton mode-locked microcomb with a repetition rate of 100 GHz, spanning a spectral range exceeding 100 nm, where more than 30 comb teeth exhibit power levels above -30 dBm at direct output. Consequently, these comb teeth can effectively serve optical computing tasks. The limitations for comb teeth with lower power primarily lie in two aspects: excessive signal-to-noise ratio, which increases noise-induced errors, deteriorates modulation stability, and reduces computational accuracy; and utilization of lower-power comb teeth deteriorate the overall comb flatness, negatively affecting subsequent amplification by EDFAs. Specifically, uneven spectral flatness can cause significant gain fluctuations in EDFAs, leading to variations in amplification efficiency and output power instability among comb lines, thus impairing the uniformity and reliability of optical computation.

To further highlight the achievable state-of-the-art comb performance, our upcoming research demonstrates significant advances, which can bring the most advanced comb light source for optical computing (arXiv preprint arXiv:2503.02022 (2025)). By introducing a passive resonant coupler, we effectively enhanced the pump power without increasing external energy consumption, which

significantly broadened the soliton microcomb bandwidth and improved its thermal robustness. By evaluating two critical metrics: the 3-dB bandwidth and the number of comb lines exceeding 100 nW, we found: With an external laser: the resonant-coupler configuration significantly broadens the spectrum, spanning from 1160 nm to 2200 nm, achieving a 3-dB bandwidth of 15.8 THz and generating 841 comb lines above 100 nW, with central comb line power around -10.9 dBm. With an integrated on-chip laser: this self-injection-locked approach yields single-soliton microcombs with a 98.8 GHz FSR, exceeding an optical bandwidth of 300 nm, producing 171 comb lines above 100 nW. This represents the broadest reported soliton microcomb at such an FSR when driven by integrated lasers.

Summarizing this point, considering only the frequency comb performance, we can achieve current state-of-the-art indicators, generating nearly a thousand comb teeth. Therefore, the comb source performance does not limit the chip scalability; instead, the crucial consideration shifts to the optical architecture's capability of fully utilizing these comb lines for computation.

(ii) The WDM devices:

Regarding the WDM performance, the selected DWDM device matches the frequency comb's 100 GHz FSR. The device we adopted provides 40 discrete channels covering wavelengths from C21 to C60, with adjacent-channel isolation exceeding 25 dB, excellently matching the comb lines used in our experiments (FS FMU-D402160M). Utilizing broader C+L-band DWDM systems can increase the number of discrete channels to hundreds. Additionally, DWDM with 50 GHz channel spacing could match a 50 GHz comb source, offering even more independent channels. Thus, the current state-of-the-art DWDM technology does not substantially limit chip scalability.

(iii) The performance of MRR:

Based on this analysis, while optical frequency combs and DWDM devices provide abundant spectral resources, the scalability of the chip ultimately depends on how effectively the MRR computing architecture can utilize and manage these resources

within practical design constraints. The single-layer MRR array size determines the achievable matrix scale, primarily constrained by the number of resonance peaks that can fit into one FSR without interference upon electrical tuning. Here, a trade-off emerges: reducing MRR radius increases FSR and accommodates more resonance peaks; however, smaller radius concurrently enhances bending losses, reduces Q-factor, and broadens resonance linewidth, thereby restricting the maximum number of non-overlapping resonance peaks. Conversely, increasing the radius reduces FSR and linewidth but excessively narrow linewidth complicates precise modulation due to limited tuning resolution of voltage-controlled modulators. Thus, the MRR radius must balance losses against integration density.

According to our current design and experimental validation, a 5 μm -radius MRR provides an optimal balance, yielding a 17.35 nm FSR. Non-overlapping resonance peaks separated by approximately 0.4 nm have been experimentally verified, achieving a high modulation precision of 5.64 bits, theoretically enabling a single-layer 43×43 MRR array, equivalent to 1849 tunable devices with 43 input and 43 output ports. A precision of 5.64 bits allows weight resolutions down to 0.01, but for lower-precision applications (e.g., image filtering and denoising requiring fewer discrete weights), narrower linewidths accommodating more peaks per FSR can further increase the array size to tens of thousands of tunable devices per layer.

Moreover, our cross-coupled waveguide-MRR architecture leverages multiple FSRs, significantly enhancing scalability through wavelength multiplexing. With the demonstrated electrical tuning spanning five FSRs, we estimate a realistic, experimentally achievable configuration can cascade four array layers while maintaining the high modulation precision of 5.64 bits, corresponding to a total of $43 \times 43 \times 4 + 43 \times (4-1) = 7525$ tunable devices, with 215 input ports and 172 output ports.

2) From the aspect of the thermo-optical tuning technology.

With only a single AlCu layer, metal traces cannot cross one another, so they must snake around previously routed lines; this greatly increases trace length and resistance,

leads to larger IR drops and RC delays, wastes valuable silicon area, and ultimately limits the integration density of reconfigurable photonic chips. AlCu itself is relatively resistive and thin in standard MPW flows, so long runs approach electromigration limits and introduce thermal and electrical crosstalk that degrades phase stability. These effects undermine computational accuracy and stability, particularly for MRRs, whose resonance is highly sensitive, forcing us to perform frequent recalibration during operation. Consequently, further scalability will require careful optimization of the wiring architecture. By refining the electrical wire routing (e.g., adopting multi-metal stacks, local cross-over bridges, or flip-chip driver ASICs), we can greatly shorten line lengths, reduce IR drop, and provide the uniform, low-loss drive necessary to support much larger computing arrays.

Supplementary Note 12 | Comparison with the state-of-the-art photonic neural networks.

Table. S4 | Comparison with the state-of-the-art photonic neural networks in the key performance.

Ref	Light source	Photonic components	Computing Speed	Area efficiency	Energy consumption*	Tunable components	Input/output dimension	Network model	Tasks performed
This work	Chip-to-chip homemade microcomb	MRR + MZI	2.64 TOPS^c	2.45 TOPS/mm^{2 c}	179W	208	32/24	1.FCNN 2.CNN 3.RNN	1. Dynamic temporal tasks 2. Static tasks
[1]	CW	Crossbar array	0.54 TOPS ^c	1.2 TOPS/mm ^{2 c}	0.107 W	27	9/3	CNN	Static tasks
[2]	CW	MZI	0.28 TOPS ^c	0.054 TOPS/mm ^{2 c}	\	84	6/6	PNN	Static tasks
[3] ^a	Chip-to-chip homemade microcomb	Crossbar array	1.2 TOPS	1.2 TOPS/mm ²	2.5 W	36	9/4	CNN	Static tasks
[4]	CW	Inverse-designed	\ ^b	\ ^b	\	\	3/3	^d	Static tasks
[5]	CW and PW	MZI	0.33 TOPS ^c	0.20 TOPS/mm ^{2 c}	45 W	132	6/6	PNN	Static tasks
[6]	CW	TDU	21 TOPS ^c	40 TOPS/mm ^{2 c}	199.88 W	80	16/4	DNN	Static tasks
[7]	CW	MZI	2.52 TOPS ^c	3.0 TOPS/mm ^{2 c}	\	91	8/8	PNN	1. Dynamic temporal tasks 2. Static tasks
[8]	CW	MZI	1310 TOPS ^c	1 TOPS/mm ^{2 c}	79.6 W	32896	12/12	1. CNN 2. MLP 3. BERT	1. Dynamic temporal tasks 2. Static tasks
[9]	CW	MZI	82 TOPS ^c	\	\	4160	4/64	^d	Static tasks
[10]	CW	MZI	0.20 TOPS ^c	\ ^b	\	0	4/4	^d	Static tasks
[11]	CW	DU + MZI	5.08 TOPS ^{b c}	0.48 TOPS/mm ^{2 b c}	260.32 W	160	64/8	PNN	1. Dynamic temporal tasks 2. Static tasks
[12]	CW	DER + DL	1.25 TOPS	0.061 TOPS/mm ²	70.43 W	144	16/16	CNN	1. Dynamic temporal tasks 2. Static tasks
[13]	CW	MRR + DL + DU	\ ^b	\ ^b	5.1 W	\	9/45	RC	1. Dynamic temporal tasks 2. Static tasks
[14]	Chip-to-chip homemade microcomb	MRR + DL	0.14 TOPS	1.1 TOPS/mm ²	89.5 W	4	1/1	CNN	Static tasks
[15]	CW	MZI + DU	0.053 TOPS ^{b c}	1.5 TOPS/mm ^{2 b c}	\	40	10/10	CNN	Static tasks
[16]	CW	MRR + DL	0.48 TOPS	0.59 TOPS/mm ²	7.96 W	36	9/9	CNN	Static tasks
[17]	CW	MZI + DU	1.36 TOPS ^c	0.26 TOPS/mm ^{2 c}	224.258 W	9	8/8	CNN	Static tasks
[18]	CW	MMI + MRR	0.93 TOPS	0.072 TOPS/mm ²	78 W	32	8/8	1. CNN 2. RNN	1. Dynamic temporal tasks

Note:

a. The Si₃N₄ platform, **b.** Area efficiency for tunable components, **c.** At a typical frequency of 10 GHz (All frequencies below 10 GHz are calculated based on 10 GHz, while frequencies above 10 GHz are calculated based on the original data in the paper), **d.** Matrix-Vector Multiplication (MVM).

DU-diffraction unit

TDU-tunable diffraction unit

PNN-photonic neural network

DNN-diffractive neural network

*The reported energy consumption varies significantly across references, as some do not specify whether electrical energy (e.g., from drivers or peripheral circuits) is included.

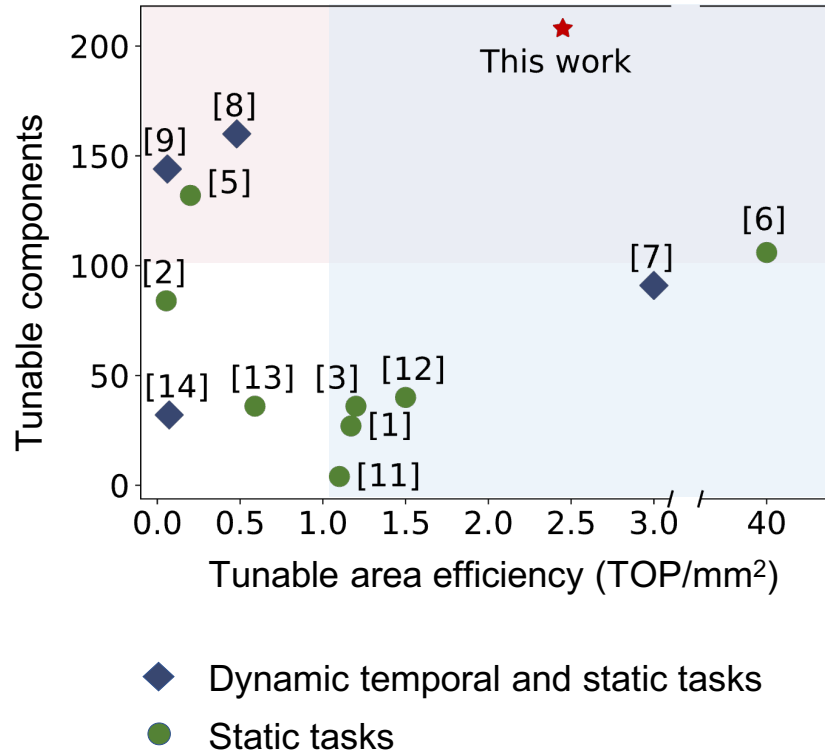


Fig. S28 | Comparison with the state-of-the-art photonic neural networks in the tunable area efficiency and neurons.

References:

1. Dong, B. et al. Partial coherence enhances parallelized photonic computing. *Nature* 632, 55–62 (2024).
2. Pai, S. et al. Experimentally realized in situ backpropagation for deep learning in photonic neural networks. *Science* 380, 398–404 (2023).
3. Feldmann, J. et al. Parallel convolutional processing using an integrated photonic tensor core. *Nature* 589, 52–58 (2021).
4. Nikkhah, V. et al. Inverse-designed low-index-contrast structures on a silicon photonics platform for vector–matrix multiplication. *Nat. Photon.* 18, 501–508 (2024).
5. Bandyopadhyay, S. et al. Single-chip photonic deep neural network with forward-only training. *Nat. Photon.* 18, 1335–1343 (2024).
6. Cheng, J. et al. Multimodal deep learning using on-chip diffractive optics with in situ training capability. *Nat Commun* 15, 6189 (2024).
7. Du, Z. et al. Ultracompact and multifunctional integrated photonic platform. *Sci. Adv.* 10, eadm7569 (2024).
8. Ahmed, S.R., Baghdadi, R., Bernadskiy, M. et al. Universal photonic artificial intelligence acceleration. *Nature* 640, 368–374 (2025).
9. Hua, S., Divita, E., Yu, S. et al. An integrated large-scale photonic accelerator with ultralow latency. *Nature* 640, 361–367 (2025).

10. Najjar Amiri, A., Vit, A.D., Gorgulu, K. et al. Deep photonic network platform enabling arbitrary and broadband optical functionality. *Nat Commun* 15, 1432 (2024).
11. Xu, Z. et al. Large-scale photonic chiplet Taichi empowers 160-TOPS/W artificial general intelligence. *Science* 384, 202–209 (2024).
12. Hong, S. et al. Versatile parallel signal processing with a scalable silicon photonic chip. *Nat Commun* 16, 288 (2025).
13. Wang, D., Nie, Y., Hu, G., Tsang, H. K. & Huang, C. Ultrafast silicon photonic reservoir computing engine delivering over 200 TOPS. *Nat Commun* 15, 10841 (2024).
14. Bai, B. et al. Microcomb-based integrated photonic processing unit. *Nat Commun* 14, 66 (2023).
15. Zhu, H. H. et al. Space-efficient optical computing with an integrated chip diffractive neural network. *Nat Commun* 13, 1044 (2022).
16. Xu, S., Wang, J., Yi, S. & Zou, W. High-order tensor flow processing using integrated photonic circuits. *Nat Commun* 13, 7970 (2022).
17. Huang, Yuyao, et al. "Diffraction - driven parallel convolution processing with integrated photonics." *Laser & Photonics Reviews* 19.4 (2025): 2400972.
18. Du, S. et al. Scalable parallel photonic processing unit for various neural network accelerations. *Photon. Res.* 12, 2691 (2024).