

Supplementary Information - Deep Learning-Driven Performance Prediction and Design of High-DoF MEMS Resonators

Zeyu Jia^{1,2}, Chaohui Wu^{1,3}, Maoyun Li¹, Wenhui Yan², Xiaoxu Wang^{2,}, Miao Lei¹, Jiaxin Sun²,*

Shuang Wu^{1,2}, Chaoyue Shi¹, Dehua Tan¹, Wukai Li², Yuhao Wang², Pengfei Li⁴, Qianbo Lu^{1,3,}*

Affiliations

¹State Key Laboratory of Flexible Electronics (LOFE) & Institute of Flexible Electronics (IFE), Northwestern Polytechnical University, 127 West Youyi Road, Xi'an, 710072, China.

²School of Automation, Northwestern Polytechnical University, 127 West Youyi Road, Beilin District, Xi'an 710072, China.

³Shaanxi Key Laboratory of Flexible Electronics, Northwestern Polytechnical University, 127 West Youyi Road, Xi'an, 710072, China.

⁴School of Marine Science and Technology, Northwestern Polytechnical University, 127 West Youyi Road, Beilin District, Xi'an, China.

Corresponding author

Correspondence to: Qianbo Lu: iamqlu@nwpu.edu.cn, Xiaoxu Wang: woyaofly1982@nwpu.edu.cn.

Supplementary Note 1

Vibration mode identification

Modal analysis enables the extraction of multiple resonant frequencies, mode shapes, participation factors, and effective modal masses of the resonator. Participation factors and modal masses are widely used as key tools for evaluating the influence of eigenvalues on state variables in dynamic systems, thereby determining the contribution or impact of specific variables on each mode. In modal analysis, the vibrational motion of the resonator can be projected into six degrees of freedom: translations along the X , Y , or Z axes (XT, YT, ZT) and rotations about the X , Y , or Z axes (XR, YR, ZR). By comparing the participation factors and modal masses corresponding to these six directions, vibration modes associated with different resonant frequencies can be efficiently identified, eliminating the need for manual inspection.

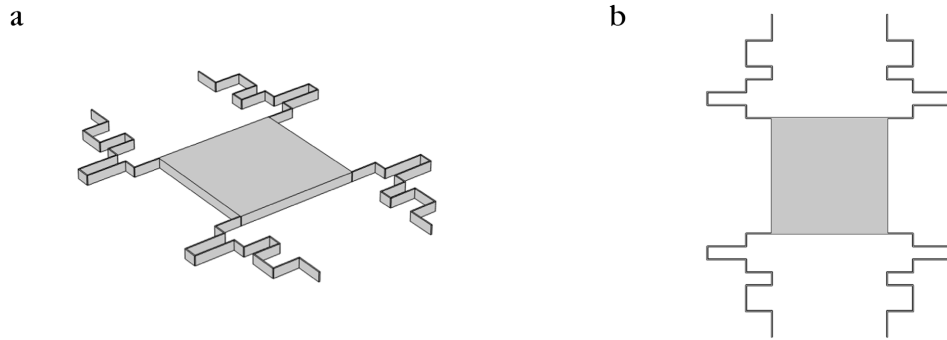


Fig. S1. Schematic of resonator structure

Using the resonator shown in Fig. S1 as an example, we demonstrate an efficient method to identify different vibration modes and extract their resonant frequencies. Table S1 lists the resonant frequencies and participation factors of each mode obtained from 3D FEM simulations, while Table S2 provides the corresponding effective modal masses. As shown in Fig. S2, the first four vibration modes of the resonator are: in-plane vibration along the X -axis (1st mode), out-of-plane vibration along the Z -axis (2nd mode), in-plane vibration along the Y -axis (3rd mode), and torsional vibration around the Y -axis (4th mode). The radar chart displays the absolute ranking of participation factors across six degrees of freedom, from the highest (rank = 1, outermost) to the lowest (rank = 6, innermost). The bar chart illustrates the magnitude of effective modal mass in each direction. For the 1st and 3rd modes (in-plane vibrations), the participation factor ranks first in the XT or YT direction, and the effective modal mass is the largest in the corresponding direction. Similarly, for the 2nd mode (out-of-plane vibration), the participation factor ranks first in the ZT direction, with the highest effective modal mass.

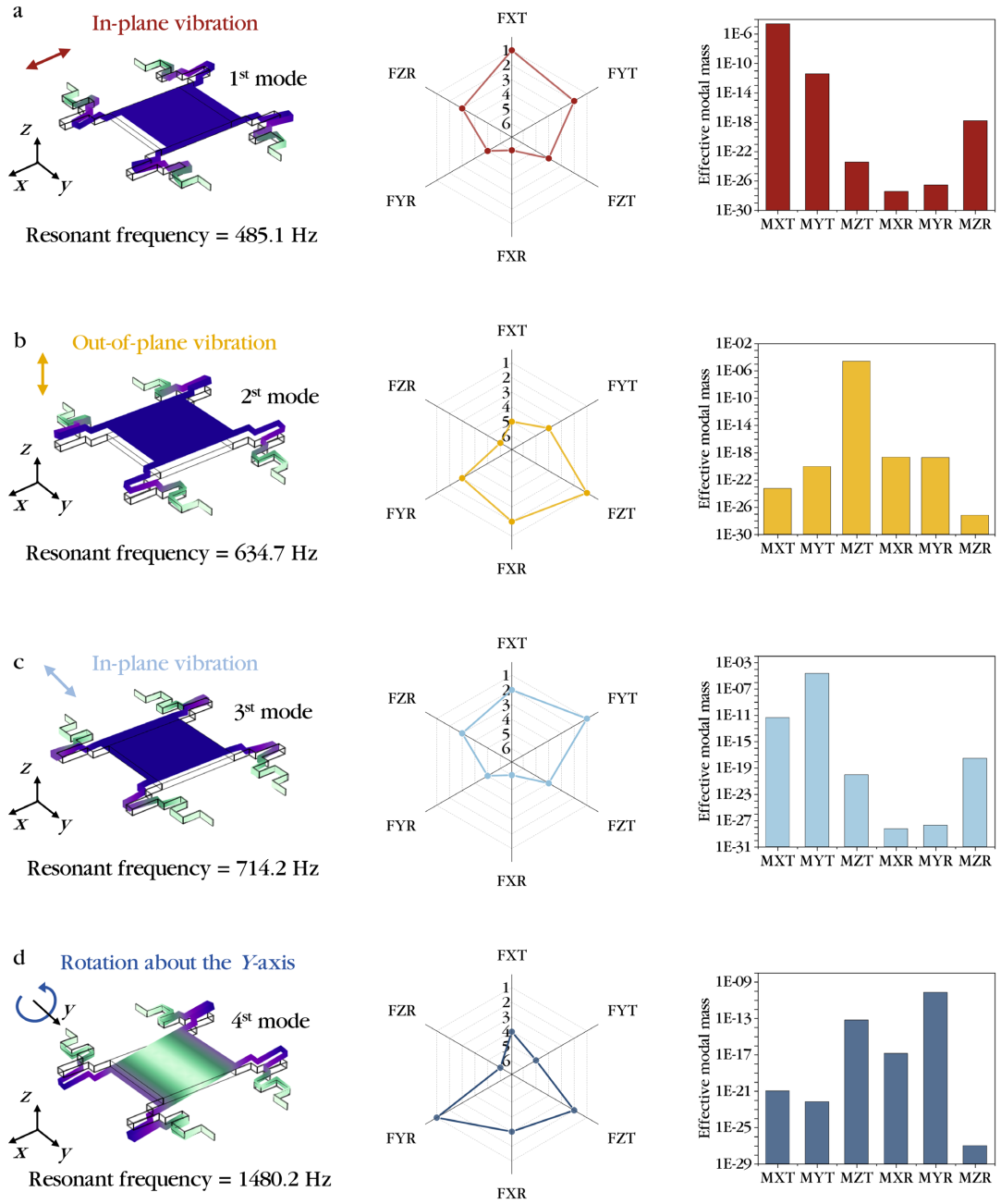


Fig. S2. Different resonant frequencies and corresponding mode shapes, along with the ranking of participation factors in different directions and distribution of effective mass.

Using this information, vibration modes of interest can be efficiently distinguished from all modes obtained through FEM analysis. By performing modal analysis and simply ranking the participation factors or effective modal masses in the six degrees of freedom, the resonant frequency of a target mode can be extracted when its participation factor or effective modal mass ranks first. The extracted resonant frequencies are then combined with the binarized structural feature vector to form the dataset for

network training. The trained ResNES model can subsequently be used to rapidly predict multi-order resonant frequencies for vibration modes of interest.

Table S1 Normalized participation factors and resonant frequencies for the first six modes.

Mode order	Resonant frequency (Hz)	Normalized participation factor, <i>X</i> -axis translation (FXT)	Normalized participation factor, <i>Y</i> -axis translation (FYT)	Normalized participation factor, <i>Z</i> -axis translation (FZT)	Normalized participation factor, rotation about the <i>X</i> -axis (FXR)	Normalized participation factor, rotation about the <i>Y</i> -axis (FYR)	Normalized participation factor, rotation about the <i>Z</i> -axis (FZR)
1	485.0	4.97E-03	1.99E-06	-1.97E-12	-1.94E-14	-5.59E-14	-1.27E-09
2	634.7	2.26E-12	-9.35E-11	4.98E-03	4.54E-10	4.26E-10	2.51E-14
3	714.0	-2.04E-06	4.93E-03	9.38E-11	-7.46E-15	-1.37E-14	-1.68E-09
4	1480.2	-3.20E-11	-8.11E-12	2.56E-07	-3.67E-09	-8.52E-06	-3.14E-14
5	1683.6	-1.10E-11	-8.99E-13	3.03E-07	-8.11E-06	3.92E-09	1.84E-13
6	1896.0	6.88E-07	8.70E-07	-1.05E-11	1.29E-13	-1.78E-14	1.13E-05

Table S2 Effective modal mass and resonant frequencies for the first six modes.

Mode order	Resonant frequency (Hz)	Effective modal mass, <i>X</i> -axis translation (FXT)	Effective modal mass, <i>Y</i> -axis translation (FYT)	Effective modal mass, <i>Z</i> -axis translation (FZT)	Effective modal mass, rotation about the <i>X</i> -axis (FXR)	Effective modal mass, rotation about the <i>Y</i> -axis (FYR)	Effective modal mass, rotation about the <i>Z</i> -axis (FZR)
1	485.0	2.47E-05	3.95E-12	3.86E-24	3.75E-28	3.13E-27	1.62E-18
2	634.7	5.12E-24	8.73E-21	2.48E-05	2.06E-19	1.81E-19	6.30E-28
3	714.0	4.17E-12	2.44E-05	8.80E-21	5.56E-29	1.88E-28	2.84E-18
4	1480.2	1.02E-21	6.57E-23	6.54E-14	1.35E-17	7.26E-11	9.89E-28
5	1683.6	1.21E-22	8.08E-25	9.16E-14	6.58E-11	1.53E-17	3.38E-26
6	1896.0	4.74E-13	7.56E-13	1.10E-22	1.67E-26	3.16E-28	1.27E-10

Supplementary Note 2

Analysis of experimental errors

Potential sources of experimental error may arise from manufacturing tolerances (e.g., minor variations in dimensions such as beam width and thickness), deviations in material properties (such as Young's modulus), and external environmental disturbances. The following analysis examines each potential source of error:

(1) Manufacturing Tolerances

Dimensional variations during fabrication represent one of the primary sources of error. Critical dimensions of MEMS resonators, such as beam width and thickness, directly induce deviations in structural stiffness and mass distribution, consequently affecting their resonant frequencies.

① Beam width variation

The beam width variation resulting from processing errors was first characterized. Using the resonator illustrated in Fig. S3 as an example, eight regions across the four elastic beams were randomly selected, and the beam widths were measured using a microscopic dimensional measurement system. As shown in Fig. S3, width measurements from 24 segments of the elastic beams indicate that processing-induced deviations were strictly constrained within 1 μm .

Since ResNES was trained based on FEM results, the physical property deviations caused by the aforementioned manufacturing tolerances were evaluated using FEM simulations. Resonant frequencies were computed via FEM for beam widths of 49 μm , 50 μm , and 51 μm . As summarized in Table S3, a beam width error of 1 μm resulted in a nearly 3% change in the first two resonant frequencies. These results demonstrate that although the beam width variation introduced by standard silicon-based microfabrication is less than 1 μm , such a minor deviation is sufficient to introduce frequency shifts on the order of several percent in the designed MEMS resonators. Therefore, fabrication-induced beam width inaccuracy can be considered a major contributor to the overall error.

Table S3 Frequency deviation of resonators with different beam widths

Beam width	First-order resonant frequency	Deviation	Second-order resonant frequency	Deviation
50 μm	249.11 Hz	-	321.24 Hz	-
49 μm	241.47 Hz	-3.0651%	311.92 Hz	-2.9026%
51 μm	256.12 Hz	2.8153%	330.72 Hz	2.9500%

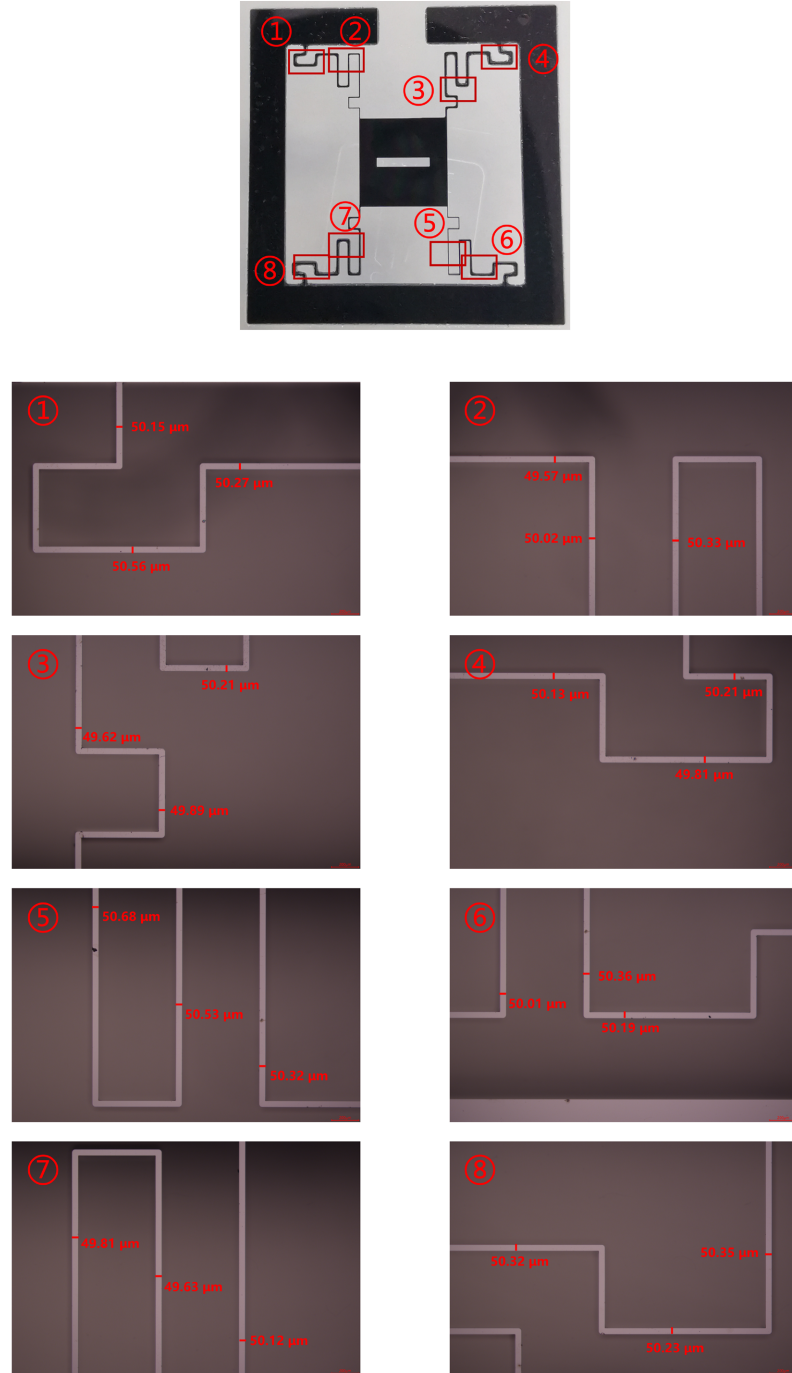


Fig. S3 Characterization of resonator beam width.

② Beam thickness variation

In the designed resonator, both the beams and the proof mass share the same thickness. Thus, thickness variations primarily originate from the wafer thickness tolerance used in fabrication. In this work, the overall thickness variation of the wafers employed for resonator fabrication was specified as $400 \mu\text{m} \pm 5 \mu\text{m}$.

Taking the resonator shown in Fig. S3 as an example, FEM simulations were performed to obtain the resonant frequencies at beam thicknesses of 395 μm , 400 μm , and 405 μm . The resulting deviations in the first two resonant frequencies due to thickness variation are summarized in Table S4. The results indicate that wafer thickness deviation has a negligible influence on resonant frequencies, with an error of approximately 0.02%. Therefore, its impact on the experimental outcomes can be considered insignificant.

Table S4 Frequency deviation of resonators with different thicknesses.

Resonator thickness	First-order resonant frequency	Deviation	Second-order resonant frequency	Deviation
400 μm	249.11 Hz	-	321.24 Hz	-
395 μm	249.05 Hz	-0.0223%	321.18 Hz	-0.0176%
405 μm	249.16 Hz	0.0203%	321.30 Hz	0.0177%

(2) Material property variations

The simulations employed ideal material parameters for single-crystal silicon (Young's modulus: 170 GPa, density: 2329 kg/m³). However, the actual 4-inch wafers used may exhibit variations in Young's modulus of up to $\pm 5\%$ due to differences in crystal orientation and doping concentration.

FEM simulations were conducted to obtain the resonant frequencies at Young's moduli of 161.5 GPa and 178.5 GPa. The resulting deviations in resonant frequencies due to these material variations are summarized in Table S5. As shown in Table S5, a 5% variation in Young's modulus leads to highly consistent deviations in the first two resonant frequencies, both exceeding 2%. These results indicate that material property variations represent a significant source of experimental error.

Table S5 Frequency deviation of resonators with different Young's moduli.

Young's modulus	First-order resonant frequency	Deviation	Second-order resonant frequency	Deviation
170.0 GPa	249.11 Hz	-	321.24 Hz	-
161.5 GPa	242.80 Hz	-2.5329%	313.11 Hz	-2.5316%
178.5 GPa	255.26 Hz	2.4679%	329.17 Hz	2.4688%

(3) External Disturbances

To minimize environmental noise, all experiments were conducted in an underground cave laboratory. Within the effective frequency band, the root mean square (RMS) value of the velocity noise was maintained below 3.16×10^{-8} m/s on average. The average ambient temperature in the laboratory was 18.2°C, with annual variations limited to less than 0.2°C and daily fluctuations below 0.03°C.

Furthermore, the resonant frequency and vibration amplitude differed significantly between the outer frame and the resonator, ensuring that outer frame-induced noise contributed negligibly to the experimental error.

To address the main error sources identified above, several practical improvement strategies can be implemented in future work. Regarding beam-width variation—the dominant fabrication-related error—a process feedforward compensation strategy can be adopted. This method pre-corrects critical dimensions during mask design using historical process data to counteract systematic etching deviations. Additionally, optimizing the deep reactive ion etching (DRIE) process parameters can enhance critical dimensions accuracy, thereby constraining beam-width variations to a narrower range. At the simulation and modeling stage, error compensation can be achieved by employing experimentally calibrated material and structural parameters in the FEM simulations. For instance, to mitigate errors stemming from material properties, nanoindentation tests can be conducted on the actual wafers, and the measured Young's modulus values can be used in place of idealized parameters, substantially reducing prediction deviations due to material-property mismatch.

Supplementary Note 3

ResNES computational efficiency and robustness evaluation.

The computational efficiency advantage of ResNES must be established through comprehensive and diverse FEM benchmarks to demonstrate its robustness. The following validation extends FEM comparisons with ResNES results across two dimensions: additional hardware platforms and different physical performance metrics.

(1) Diverse Hardware Platforms

Hardware capabilities significantly impact the time cost of FEM simulations. Additional FEM analyses were conducted on three distinct hardware platforms with notable performance differences:

- A high-performance workstation (CPU: Intel Xeon Gold 6254)
- A mainstream commercial desktop (CPU: Intel Core i7-13700)
- A laptop (CPU: Intel Core i7-12700H)

Table S6 Comparison of ResNES and FEM computational efficiency for different hardware platforms.

Hardware platform	Average prediction time per 10000 samples	Average prediction time per single sample
Workstation	1323 min	7.94 s
Desktop	1770 min	10.62 s
Laptop	2146 min	12.88 s
ResNES	1.42 min	0.00852 s

ResNES was deployed on these three platforms to evaluate resonant frequencies for 10,000 samples generated from an 8×8 grid. All FEM analyses in this work were performed on the high-performance workstation, requiring 7.94 seconds per simulation, 900 to 3000 times slower than ResNES. When using more common commercial desktops or laptops, ResNES's computational efficiency advantage becomes even more pronounced, being approximately 1200 and 1500 times faster, respectively. While employing more advanced hardware platforms might marginally reduce ResNES's comparative advantage, ResNES consistently maintains a computational speedup of over three orders of magnitude compared to FEM on most mainstream computing platforms.

(2) Diverse Physical Performance Metrics

Additional physical performance metrics were used to validate the robustness of ResNES computational efficiency. Our primary research focus is to establish a ResNES-based structural design framework, where the dataset includes binarized geometric topologies and FEM computation results. The research objective of this work is the resonant frequency, which, as a typical FEM computation result, is used to demonstrate the entire process. In practice, the proposed deep learning-driven MEMS design framework is also applicable to other typical physical performance, such as effective modal mass, maximum displacement, and maximum stress. Below, we construct the ResNES models for different physical performance metrics and compare the computational efficiency of ResNES with that of FEM.

For ResNES, prediction time remains relatively consistent regardless of structural complexity. In contrast, FEM requires significantly denser meshes for more complex structures within the same design space, leading to substantially increased computational costs. For example, the average time for FEM analysis of a structure built from an 8×8 grid is only 8 seconds, whereas analyzing a 24×24 grid structure takes 43 seconds-5 times longer. Therefore, to thoroughly explore the lower bound of the computational efficiency improvement offered by ResNES, all validation studies were conducted using structures constructed from an 8×8 grid, which enables faster FEM analysis for comparative benchmarking.

FEM models were constructed to obtain both the maximum displacement and maximum stress under one gravitational acceleration, as shown in Fig. S4. The four elastic beams of the resonator were fixed at their endpoints, and a 1-g acceleration was applied to the proof mass. The resulting physical properties were combined with binarized geometric representations of the resonators to form datasets for training ResNES. The dataset consisted of 9,000 training samples and 1,000 test samples.

As shown in Fig. S5 and Fig. S6, ResNES achieved strong agreement with FEM results, with linear regression errors generally below 5%. Table S7 compares the computational time between ResNES and FEM for predicting these physical properties. ResNES demonstrated computational efficiency over 800 times faster than FEM for both maximum displacement and maximum stress, a value slightly less than three orders of magnitude. This can be attributed to the fact that FEM static analysis incurs lower computational time costs compared to modal analysis, with a single static FEM simulation taking only 6 seconds. In contrast, ResNES inference time remains largely unaffected by the type of physical output. The computational advantage of ResNES becomes more pronounced with increasing model complexity. For instance, when the FEM mesh size was reduced by 30%. Under these conditions, the computational

efficiency of the ResNES reaches 5000 times that of the FEM. For low-cost FEM computations (within 10 seconds), the computational efficiency of the ResNES exceeds 4000 times that of the FEM. For medium to high-cost FEM computations, the computational efficiency of the ResNES can readily surpass three orders of magnitude compared to the FEM.

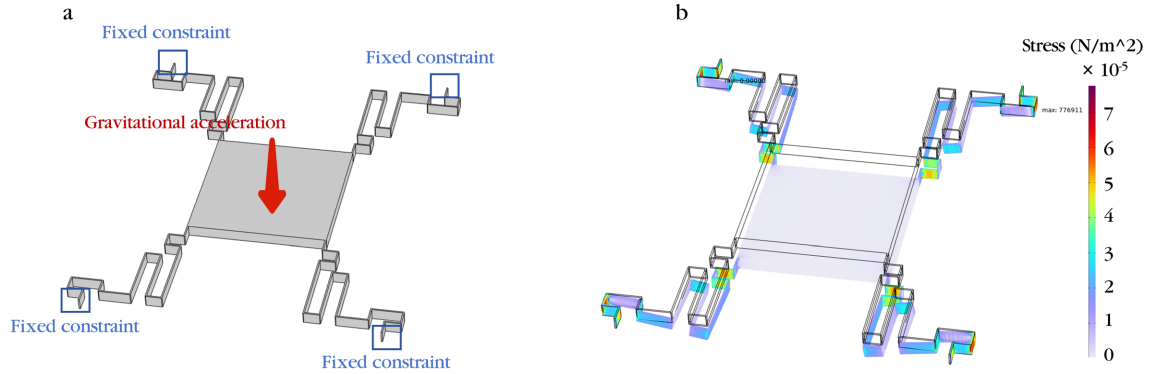


Fig. S4. Resonator's static analysis model. a Simulation model of the example resonator. b Static analysis results.

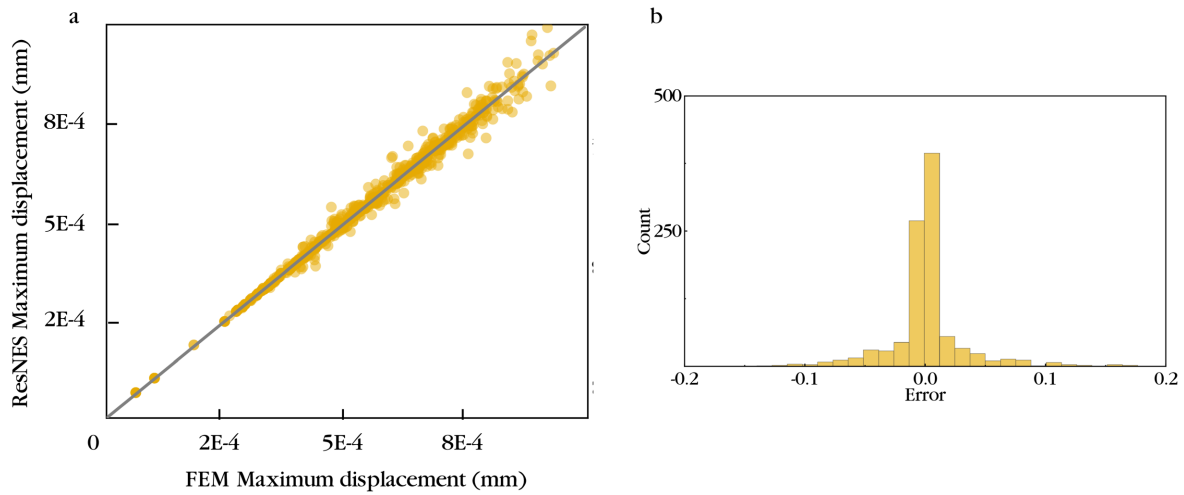


Fig. S5. Linear regression analysis of ResNES for max displacement prediction. a Linear regression between ResNES and FEM. b Linear regression error.

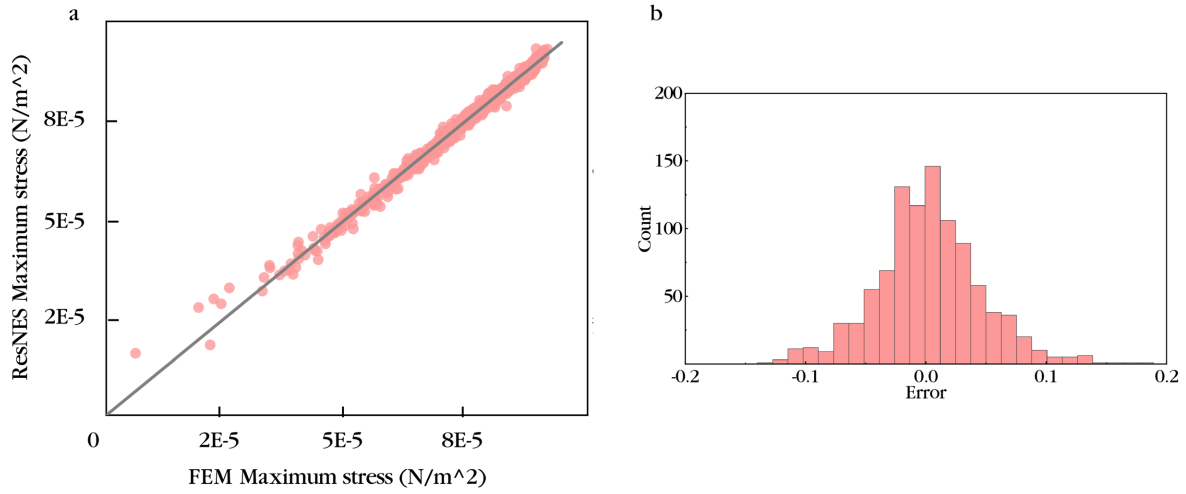


Fig. S6. Linear regression analysis of ResNES for maximum stress prediction. a Linear regression between ResNES and FEM. b Linear regression error.

Table S7 Comparison of ResNES and 3D FEM computational efficiency for different physical performance metrics.

Physical performance metrics	Average prediction time per 10000 samples (min)		Average prediction time per single sample (s)	
	ResNES	FEM	ResNES	FEM
Resonant frequency	1.42 (± 0.10)	5293.44 (± 83)	0.00852 (± 0.0006)	8.01 (± 0.5)
Maximum displacement	1.35 (± 0.10)	4413.28 (± 55)	0.00811 (± 0.0005)	6.62 (± 0.5)
Maximum stress	1.36 (± 0.10)	4484.84 (± 60)	0.00816 (± 0.0005)	6.72 (± 0.5)

In this study, the optimization process takes the maximization of resonant frequency spacing in in-plane vibration modes as an example to demonstrate the capability of ResNES as a high-performance solver integrated with optimization algorithms for rapid physical performance exploration. To further illustrate the high adaptability and customizability of the structural design framework based on ResNES, additional analyses incorporating non-operating vibration mode into the optimization objectives have been conducted. The specific optimization setup is as follows:

The geometric topology generation of resonators followed the strategy described in the “Resonator Structure Design Based on ResNES” section of the main text. Two methods were adopted for selecting resonant frequencies. The first method, as described in the main text, selects the first two resonant

frequencies of the operating vibration mode (in-plane vibration mode), denoted as f_1 and f_2 . The second method selects the f_1 together with the subsequent resonant frequency of a non-operating mode, denoted as f_{no} . The corresponding objective functions for these two frequency selection methods, f_{obj1} and f_{obj2} , can be expressed as follows:

$$f_{obj1} = 1 - \psi_1 = 1 - \frac{f_2 - f_1}{f_2 + f_1} \quad (1)$$

$$f_{obj2} = 1 - \psi_2 = 1 - \frac{f_{no} - f_1}{f_{no} + f_1} \quad (2)$$

Where, ψ_1 represents the spacing coefficient between the first two resonant frequencies of the operating vibration mode. A value closer to 1 indicates a larger frequency spacing between these two resonant frequencies corresponding to the operating mode. ψ_2 denotes the spacing coefficient between the first resonant frequency of the operating vibration mode and the resonant frequency of its adjacent non-operating vibration mode. Similarly, a value closer to 1 reflects a larger frequency spacing between these two frequencies. Based on the two optimization objectives, the optimization results obtained using the AELPSO with ResNES strategy are summarized in Table S8.

Table S8. Analysis of physical performance for resonator structures obtained under different optimization objectives.

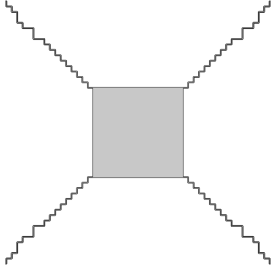
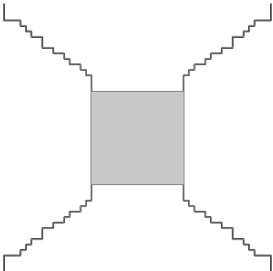
Selected Mode	Objective function	Designed geometric topology	ψ_1	ψ_2	f_1 (Hz)	f_2 (Hz)	f_{no} (Hz)
Operating vibration mode	f_{obj1}		0.597	0.428	997.01 Hz	3947.32 Hz	2491.34 Hz
Operating vibration mode + Non-operating vibration mode	f_{obj2}		0.558	0.548	607.75 Hz	2142.32 Hz	2083.61 Hz

Table S8 systematically compares the optimization results, revealing a clear trade-off in resonator performance under different optimization objectives. When the optimization specifically targets the frequency spacing between the first two in-plane vibration modes (metric ψ_1), the corresponding resonator design performs well on this metric ($\psi_1 = 0.597$). However, it also exhibits noticeable interference from spurious modes on the first operational mode, as reflected by the lower metric ψ_2 value of 0.428, shown in Fig. S7a. In contrast, by adopting the $f_{\text{obj}2}$ as the optimization function, the vibrational interference from non-operating modes is effectively suppressed, leading to a significant improvement in the anti-interference capability of the first operating mode (ψ_2 rising to 0.548). However, due to the inherent coupling among different performance metrics in multi-objective optimization, the original core optimization metric ψ_1 exhibits a reasonable decrease to 0.558, as illustrated in Fig. S7b.

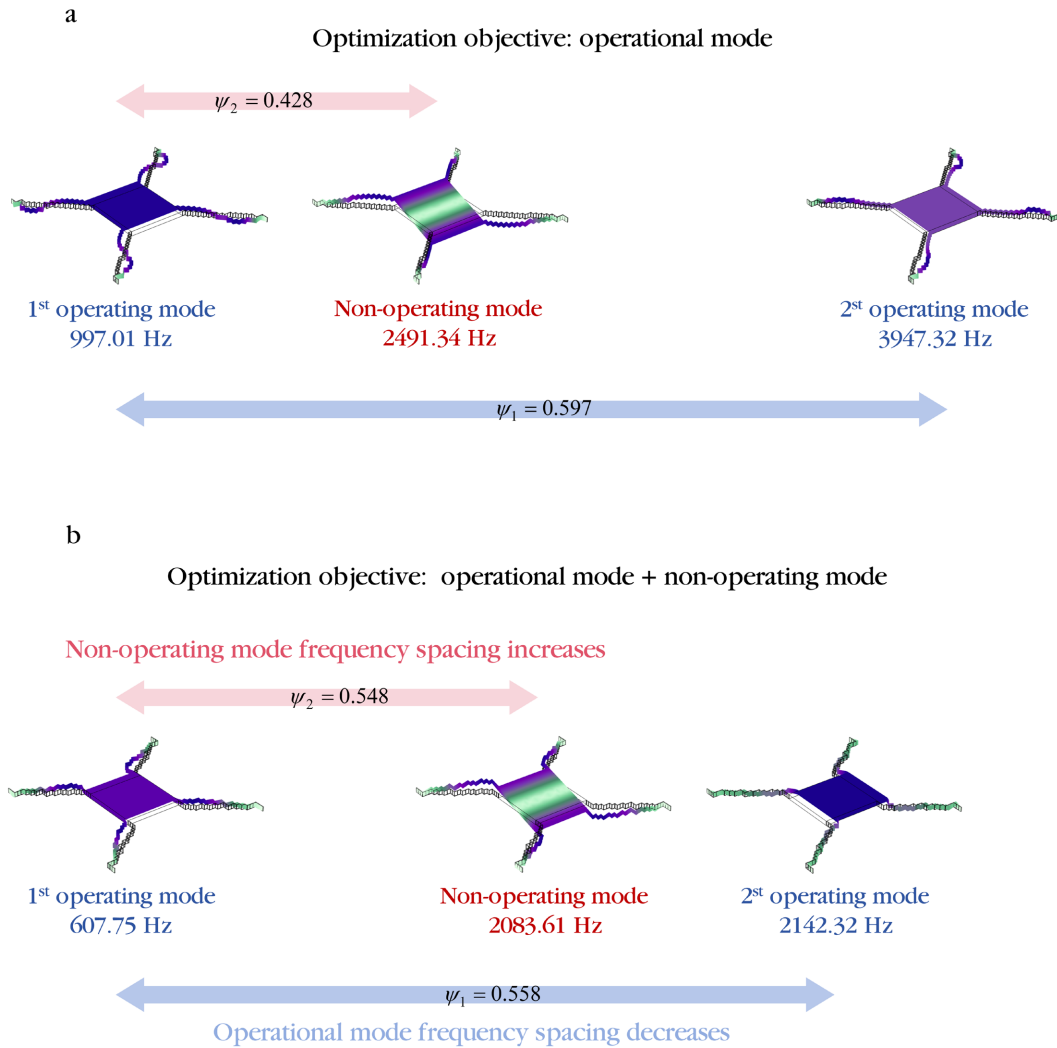


Fig. S7. Analysis of different optimization results. a Considering only the resonant frequency spacing of the

operating vibration mode. b Considering the resonant frequency spacing between the operating vibration mode and non-operating vibration modes.

The results clearly demonstrate that the ResNES possesses high adaptability and customizability, enabling the flexible customization and weighting of optimization objectives according to specific application requirements, thereby facilitating the achievement of targeted optimal designs in practice.

Supplementary Note 4

Time cost analysis of structural design based on the ResNES

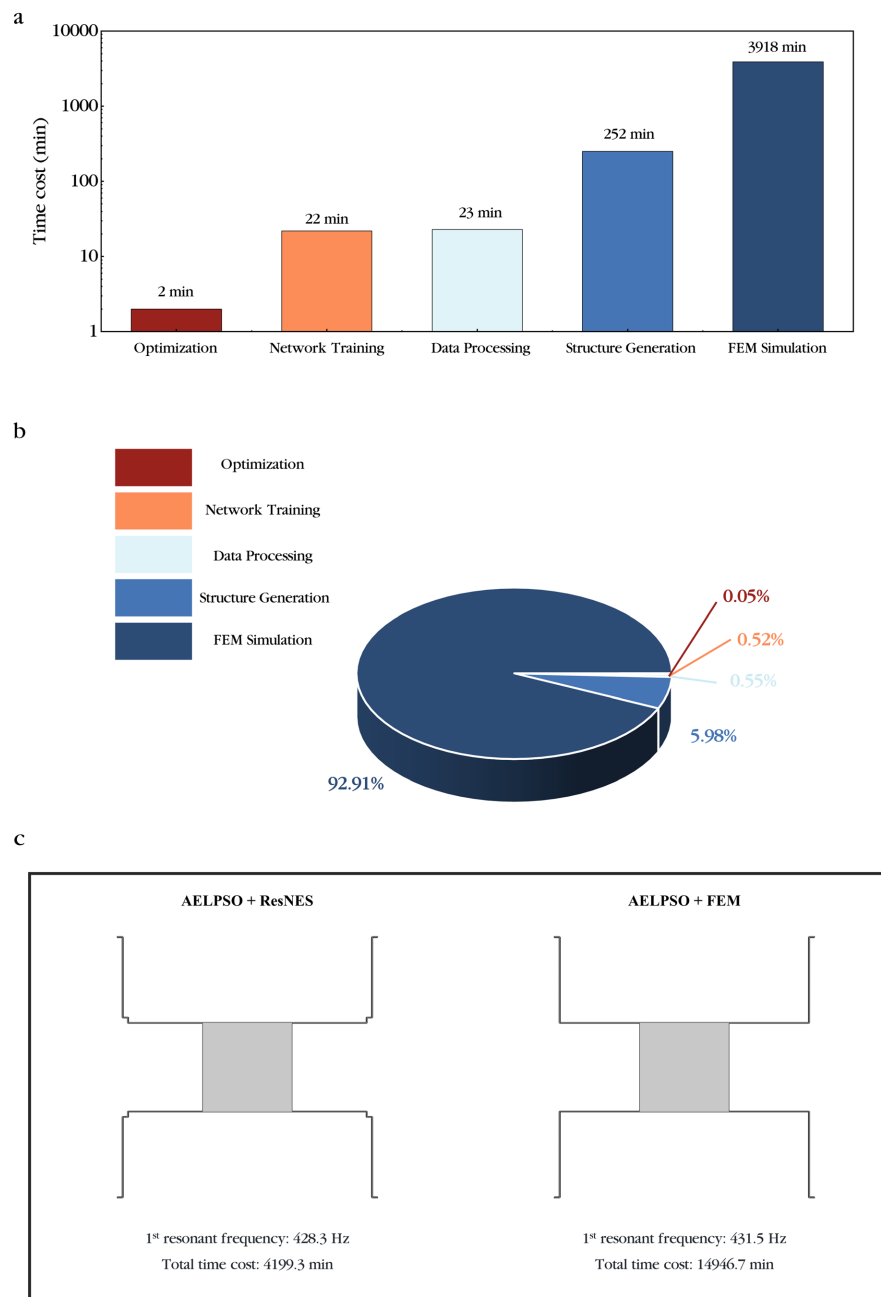


Fig. S8 Structural design time cost analysis. a. Time cost of the main processes in the structural design based on the collaborative optimization scheme. b. Percentage of time cost for each main process in the structural design based on the collaborative optimization scheme. c. Structural design results and time cost of the collaborative optimization scheme and FEM-based ranking method.

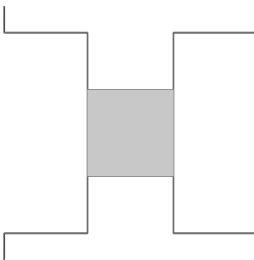
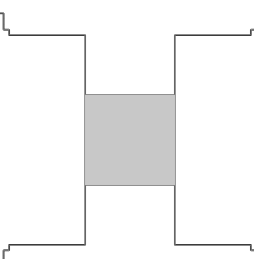
This section systematically compares the total time cost between the collaborative optimization scheme (AELPSO + ResNES) and the FEM-based ranking method, using resonant frequency

minimization as an example. Initially, 500,000 randomly generated structures with 16×16 grids were analyzed using FEM to evaluate their physical performance, followed by ranking to identify the optimal design. Subsequently, the collaborative optimization scheme was employed to refine the initial design until achieving comparable physical performance to the optimal structure obtained through ranking. The AELPSO iteration count was set to 52 after testing. The total time cost of the collaborative optimization scheme comprises several components: dataset construction (including structure generation, FEM simulation, and data processing) with a sample size of 15,000, neural network training, and optimization. The entire design process required 4,199 minutes, with the time distribution of major workflows shown in Fig. S8a. FEM simulation accounted for the largest portion of the time cost, followed by random structure generation. Notably, ResNES construction dominated the design process, while the actual optimization using ResNES constituted merely 0.05% of the total time cost, as shown in Fig. S8b. As demonstrated in Fig. S8c, while both the collaborative optimization scheme and ranking method yielded designs with similar physical properties, their time costs differed substantially. The ranking method required 14,946.7 minutes, whereas the collaborative optimization scheme, including ResNES construction, totaled only 4,199.3 minutes, achieving a 72% reduction in the overall design cycle. Importantly, the trained ResNES can be used to predict and optimize multiple physical performances, whereas the FEM-based method can only address a single physical performance. If the trained ResNES is directly employed for design, the collaborative optimization scheme can achieve a design speed improvement of more than three orders of magnitude compared to the FEM-based method.

Supplementary Note 5

Performance Comparison

Table S9. Performance Benchmarking Table1.

Source	Optimization schemes	Designed geometric topology	First resonant frequency (Hz)	Second resonant frequency (Hz)	Frequency spacing coefficient	Total Computation Time
This work	AELPSO with ML solver		409.98	932.12	0.3885	801s
Representative work ^[3]	Ranking method with ML solver		421.75	948.99	0.3846	4440s

To evaluate the advantages of the proposed structural design framework based on ResNES over existing design methods, a representative machine learning-driven MEMS design framework ^[3] was selected as a reference benchmark. We applied the geometric topology generation strategy described in the “Resonator Structure Design Based on ResNES” section of this work and adopted a multi-objective optimization scenario that simultaneously minimizes the fundamental resonant frequency and maximizes the frequency separation between the first two resonant frequencies. Table S9 provides a detailed comparison between the two frameworks in terms of key physical performance metrics and total design cycle duration. Based on the performance reports of the machine learning-driven solvers published by the reference group, it can be inferred that their solvers exhibit performance comparable to the ResNES approach proposed in this study ^[3]. The advantage of the present work lies in the adoption of a more advanced optimization algorithm. Analysis indicates that the optimization strategy combining

AELPSO and ResNES is expected to reduce the design cycle by more than fivefold while achieving comparable performance metrics.

Furthermore, compared to existing machine learning-enabled MEMS design approaches, this work demonstrates significant advantages in terms of geometric degrees of freedom, experimental validation, and optimization capabilities. We have systematically compared key design steps in our proposed machine learning-assisted framework with those of representative prior works. As shown in Table S10, the present framework seamlessly integrates high degrees of freedom structural topology generation, rapid performance prediction, efficient structural optimization, and fabrication readiness, demonstrating clear advantages over existing methods.

Table S10 Performance Benchmarking Table2.

Existing studies	Geometric topology	Machine learning model	Experimental verification	Integration with optimization algorithms
This work (2025)	High-DoF topology	Residual neural network (ResNet)	√	√
Wang et al. (2024) ^[1]	Fixed topology (disk resonator)	Dense convolutional network (DenseNet)	×	×
Huang et al. (2024) ^[2]	Fixed topology (cross-beams)	Generative adversarial network (GAN)	√	×
Lin et al. (2022) ^[3]	High-DoF topology	Residual neural network (ResNet)	×	×
Janaideh et al. (2021) ^[4]	Fixed topology (serpentine beam)	Deep neural network (DNN)	√	√

Supplementary Note 6

Activation function comparison

To further validate the effectiveness of the Leaky ReLU, we conducted additional comparative experiments. Taking samples generated from a 24×24 grid as an example, Fig. S9 compares the accuracy of the solver in predicting the first-order resonant frequency when using Leaky ReLU versus the standard ReLU. The results demonstrate that across different training sample sizes, replacing the activation function with Leaky ReLU improved the solver's accuracy by 4.12%, 1.90%, and 0.82%, respectively. This confirms that adopting Leaky ReLU effectively enhances the model's adaptability to high-dimensional feature data.

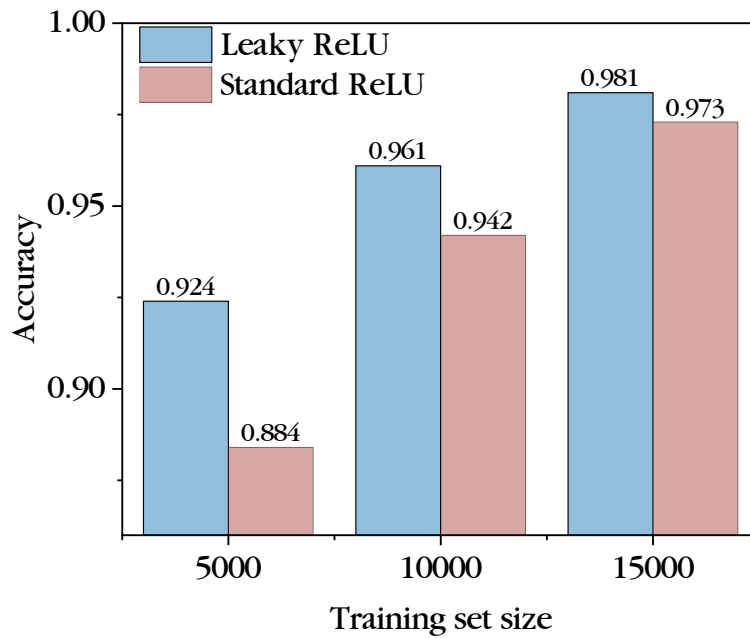


Fig. S9. Comparative analysis of computational accuracy across activation functions.

Supplementary Note 7

Machine Learning Model Comparison.

In this study, the ResNES is obtained through comprehensive training of a customized residual neural network. A thorough comparison is conducted between ResNES and two distinct machine learning models, namely DenseNet^[5] and TabTransformer^[6]. The evaluation is primarily performed from two perspectives: prediction accuracy and computational time cost. The datasets consisting of 10,000 samples generated from resonator structures with 16×16 and 24×24 grids are constructed for assessment. The dataset is split into training and testing sets in an 8:2 ratio, comprising 8,000 and 2,000 samples, respectively. To ensure fairness and reproducibility, all models are trained on an NVIDIA V100 GPU for a fixed number of 200 epochs.

① Computational Accuracy

As shown in Fig. S10, for samples generated using 16×16 grids, the prediction accuracy of solvers based on the two machine learning models is comparable to that of ResNES. Both demonstrate good agreement with FEM results, with data points distributed evenly along the $y = x$ line. However, for structures with 24×24 grids, the prediction error of ResNES is significantly lower than that of the other models, as illustrated in Fig. S11. As summarized in Table S11, the average prediction error of ResNES remains below 5%, outperforming both alternative solvers. In contrast, DenseNet exhibits a substantial increase in error when processing high-dimensional data (24×24 grid), whereas TabTransformer achieves accuracy closest to that of ResNES. In summary, in terms of computational accuracy, ResNES-constructed using a customized residual neural network-demonstrates superior performance over the other two machine learning models.

Table S11 Comparison of computational accuracy of solvers constructed with different machine learning models

Grid density	Machine learning model	Error
8×8	DenseNet	0.98231
	TabTransformer	0.99473
	Customized residual neural network	0.99735
16×16	DenseNet	0.95021
	TabTransformer	0.97153
	Customized residual neural network	0.9794
24×24	DenseNet	0.90213
	TabTransformer	0.95833
	Customized residual neural network	0.96083

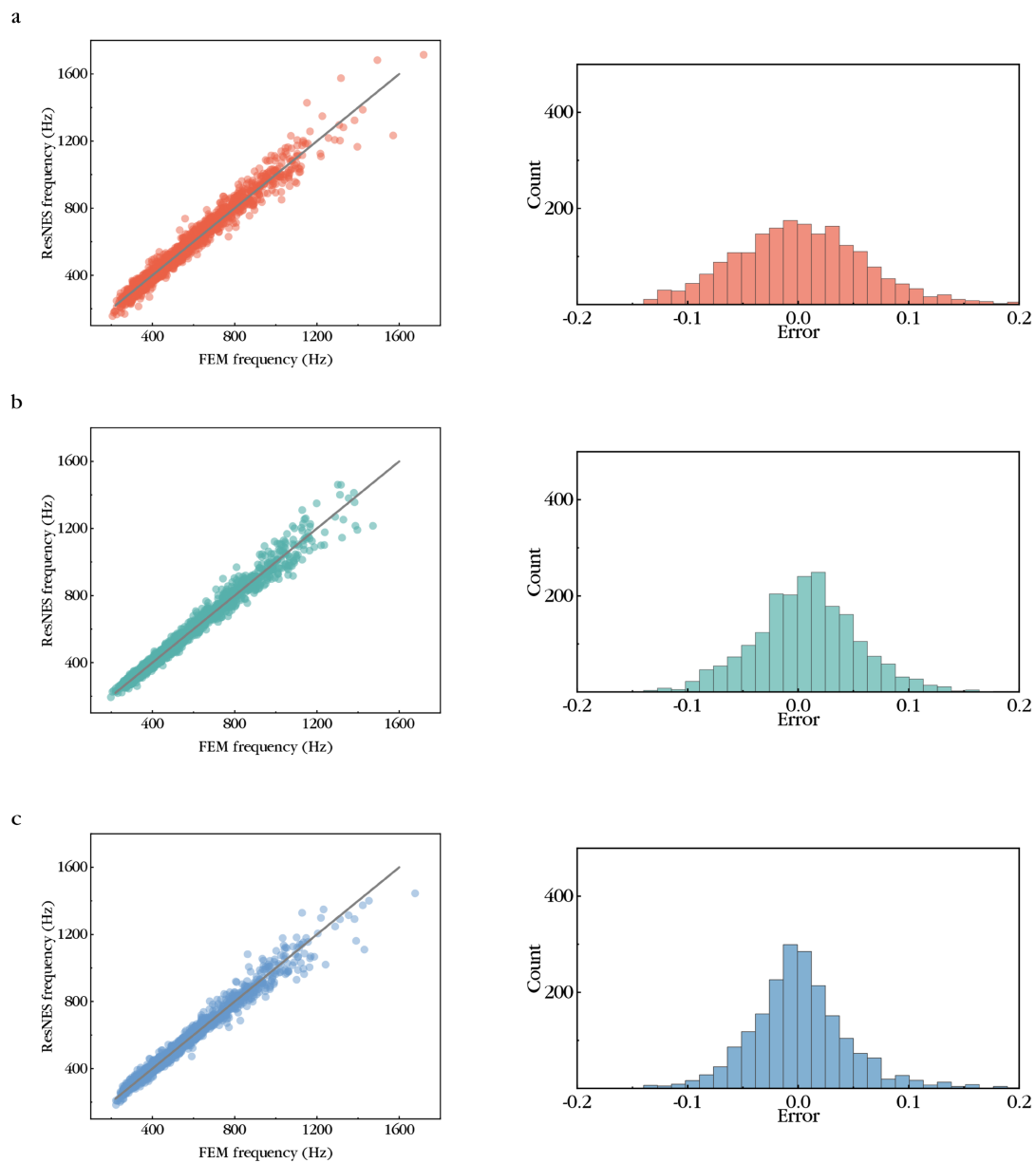


Fig. S10. Linear regression analysis of solver predictions constructed with different machine learning models versus FEM (samples generated from 16×16 grid). a DenseNet. b TabTransformer. c Customized residual neural network.

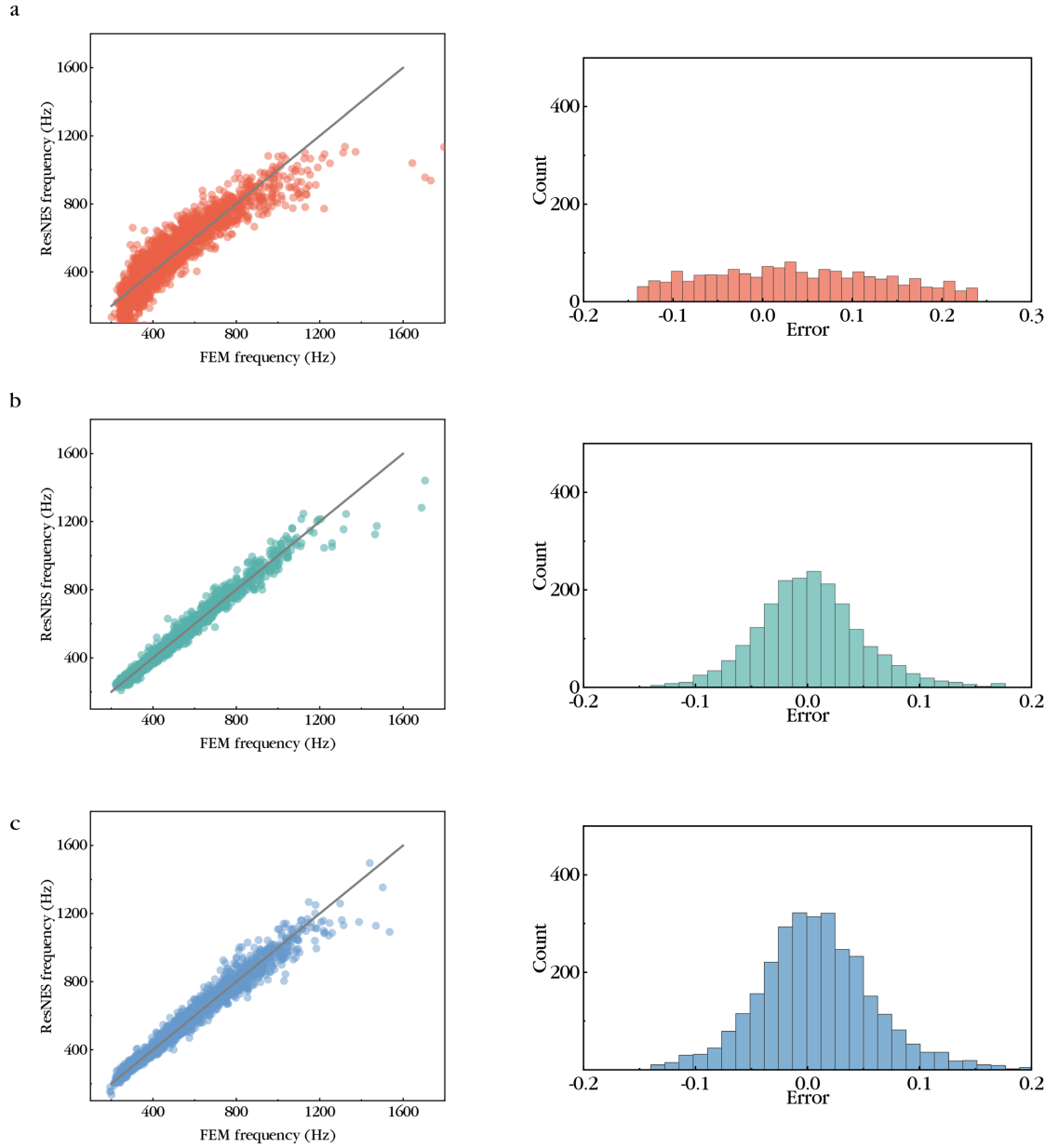


Fig. S11. Linear regression analysis of solver predictions constructed with different machine learning models versus FEM (samples generated from 24×24 grid). a) DenseNet. b) TabTransformer. c) Customized residual neural network.

② Computational Time Cost

Furthermore, taking samples constructed with 16×16 grids as an example, the training and prediction time costs of solvers empowered by different machine learning models are systematically evaluated. As summarized in Table S12, although the training time varies among different machine learning models, the major portion of the time cost for all models lies in dataset preparation. This is attributed to the fact that the dataset is generated based on FEM simulations, which are computationally

intensive. Moreover, the prediction times of solvers constructed with different machine learning models are relatively close, with differences within 5%.

Table S12 Comparative analysis of time costs across machine learning models

Machine learning model	Solver construction time			Solver prediction time
	Dataset preparation time	Average epoch time / Total epoch time	Total computation time	Average prediction time per 10000 samples
DenseNet	2930 min	7.87 s / 26.2 min	2956.2 min	1.58 min
TabTransformer	2930 min	8.12 s / 27.1 min	2957.1 min	1.62 min
Customized residual neural network	2930 min	6.82 s / 22.7 min	2952.7 min	1.69 min

In summary, the structural design framework proposed in this work is compatible with other machine learning models. Compared to existing machine learning models, the customized residual neural network adopted in our study demonstrates competitive computational accuracy while maintaining comparable computational time costs during both model training and prediction phases.

Supplementary Note 8

Dataset expansion

(1) Diverse Geometric Features

The core concept of constructing resonator elastic beams in this work involves decomposing the entire beam into multiple segments, where the two endpoints of each segment are represented as "1". To concisely illustrate the modeling approach, this study assumes that after generating one endpoint, the next endpoint can only be generated in three potential directions, with straight lines connecting consecutive endpoints. These constraints result in orthogonal beams with 90° angles. In practice, specific strategies or constraints can be introduced to generate elastic beams with non-orthogonal segments, curved geometries, or annular features.

① Non-Orthogonal Beams

For non-orthogonal beams, the three original potential directions can be expanded to seven. As shown in Fig. S12, the potential directions for the newly generated $(N+1)$ th point in the main text are limited to left, up, and right. When the potential directions are extended, the $(N+1)$ th point can be generated in seven directions: lower-left, left, upper-left, up, upper-right, right, and lower-right. This strategy enables the random generation of geometric topologies with more complex segment angles. Fig. S13 demonstrates resonators with non-orthogonal segmented beams generated using this strategy and their corresponding binary matrices.

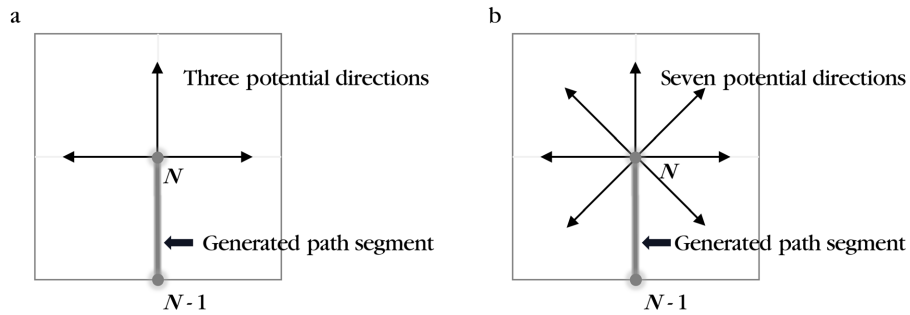


Fig. S12. Schematic of potential generation directions for the new endpoint. (a) Three potential generation directions. (b) Seven potential generation directions.

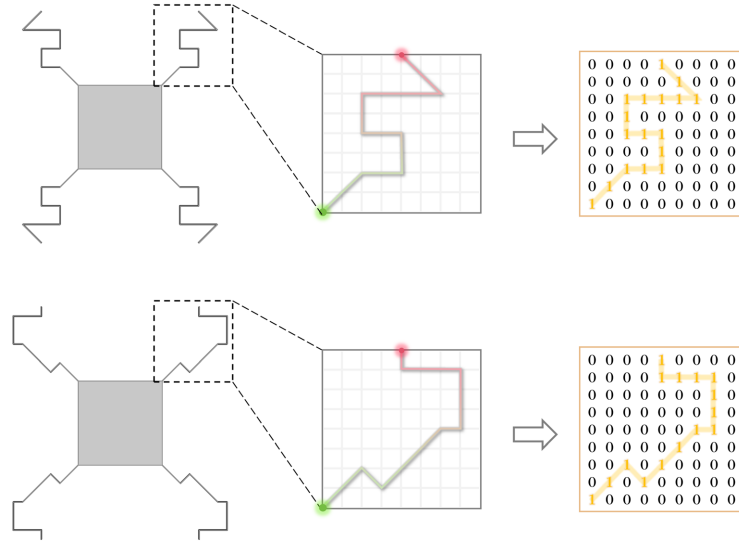


Fig. S13. Resonators with non-orthogonal beams and the corresponding binary matrices.

② Curved Beams

For the generation of curved beams, straight-line connections between endpoints can be replaced with interpolated curves to fit each segment between two endpoints, resulting in smoother and more continuous elastic beams. Fig. S14 illustrates the resonators constructed using interpolated curves to connect each endpoint, along with its corresponding binary matrix. The elastic beams generated with this method exhibit distinct curved characteristics.

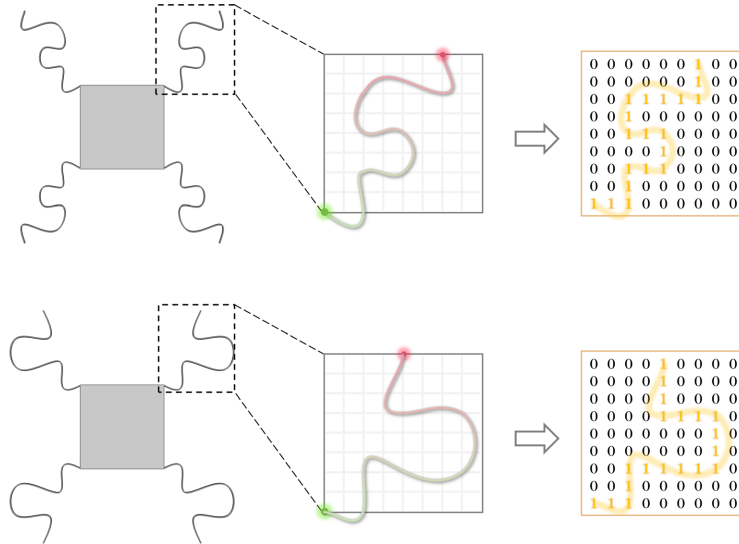


Fig. S14. Resonator with curved beams and corresponding binary matrixes.

③ Annular Features

For resonators with annular features, take the disk resonator shown in Fig. S15a as an example.

The entire disk resonator can be regarded as composed of multiple building blocks, as shown in Fig. S15b. A single building block can be obtained by radially scaling and bending a geometrically defined topology constructed within a single grid, as illustrated in Fig. S15c. The complete disk resonator is formed by repeatedly mirroring and stacking individual building blocks. Thus, the geometric features of a resonator containing annular structures can be represented using a binary matrix.

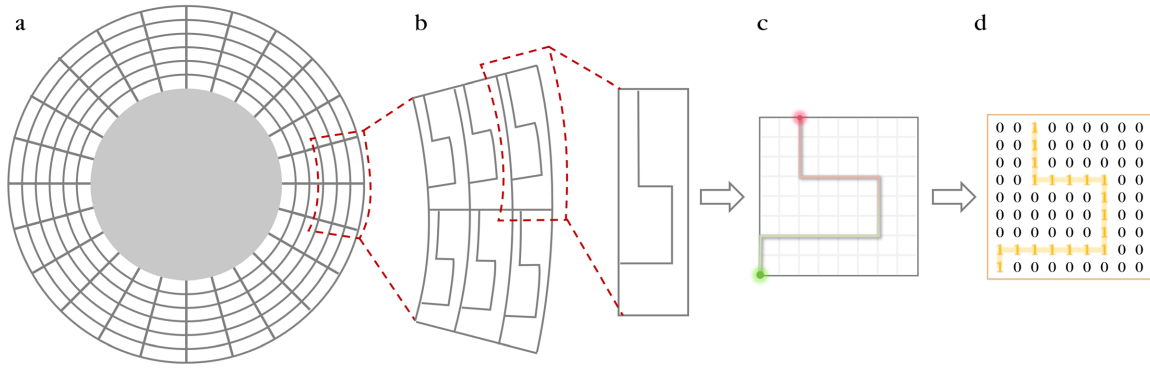


Fig. S15. Disk resonator and corresponding binary matrix. a Disk resonator. b Building block in disk resonator. c Grid and path for generating building blocks. d Binarized matrix of building blocks.

The proposed framework can be adapted to the performance analysis and design of resonators with diverse geometric features by employing different structural layouts and generation strategies. It should be noted that during the construction of ResNES, a deterministic geometric topology generation strategy is generally required to ensure that each binary matrix corresponds to a unique geometric topology.

(2) Varied Material Distributions

Fig. S16a illustrates a resonator with a specific geometric topology. It is assumed that the elastic beam of this resonator exhibits a spatially varying material distribution, consisting of two materials: silicon (Si) and silicon dioxide (SiO_2). Following the modeling approach described in the main text, the elastic beam can be discretized into multiple straight path segments. These segments are sequentially numbered from the starting point to the endpoint. As shown in Fig. S16a, the entire elastic beam is divided into five path segments.

A 5-dimensional binary column vector is introduced, where "0" represents silicon (Si) and "1" represents SiO_2 , as shown in Fig. S16b. By assigning the material corresponding to the i -th element of the column vector to the i -th path segment, a resonator with a customized material distribution can be constructed. Different material combinations can be achieved by modifying this binary vector. Fig. S16c displays three such vectors and their corresponding material distributions.

The FEM is employed to obtain physical performance metrics of the resonator under various material combinations. These metrics, together with the binary vectors, form the dataset used for training the neural network. The resulting ResNES can be utilized to rapidly predict the physical performance of resonators with diverse material distributions and layouts. Furthermore, it can be integrated with optimization algorithms to identify the optimal spatial material distribution. In this example, the elastic beam is discretized into five path segments. During network training, the beam can be further subdivided into smaller segments to generate a sufficiently large number of samples.

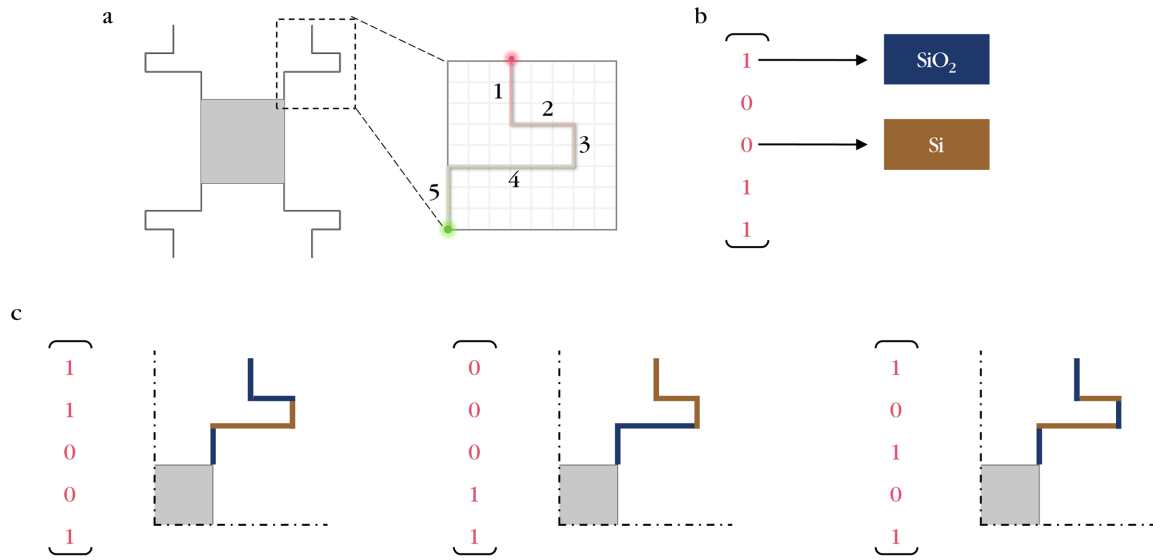


Fig. S16. Spatial material distribution design. a Resonator generated from 8×8 grid. b Column vector representing the material distribution of path segments. c Schematic of material distributions corresponding to different column vectors.

Supplementary Note 9

Comparison between 2D and 3D FEM results

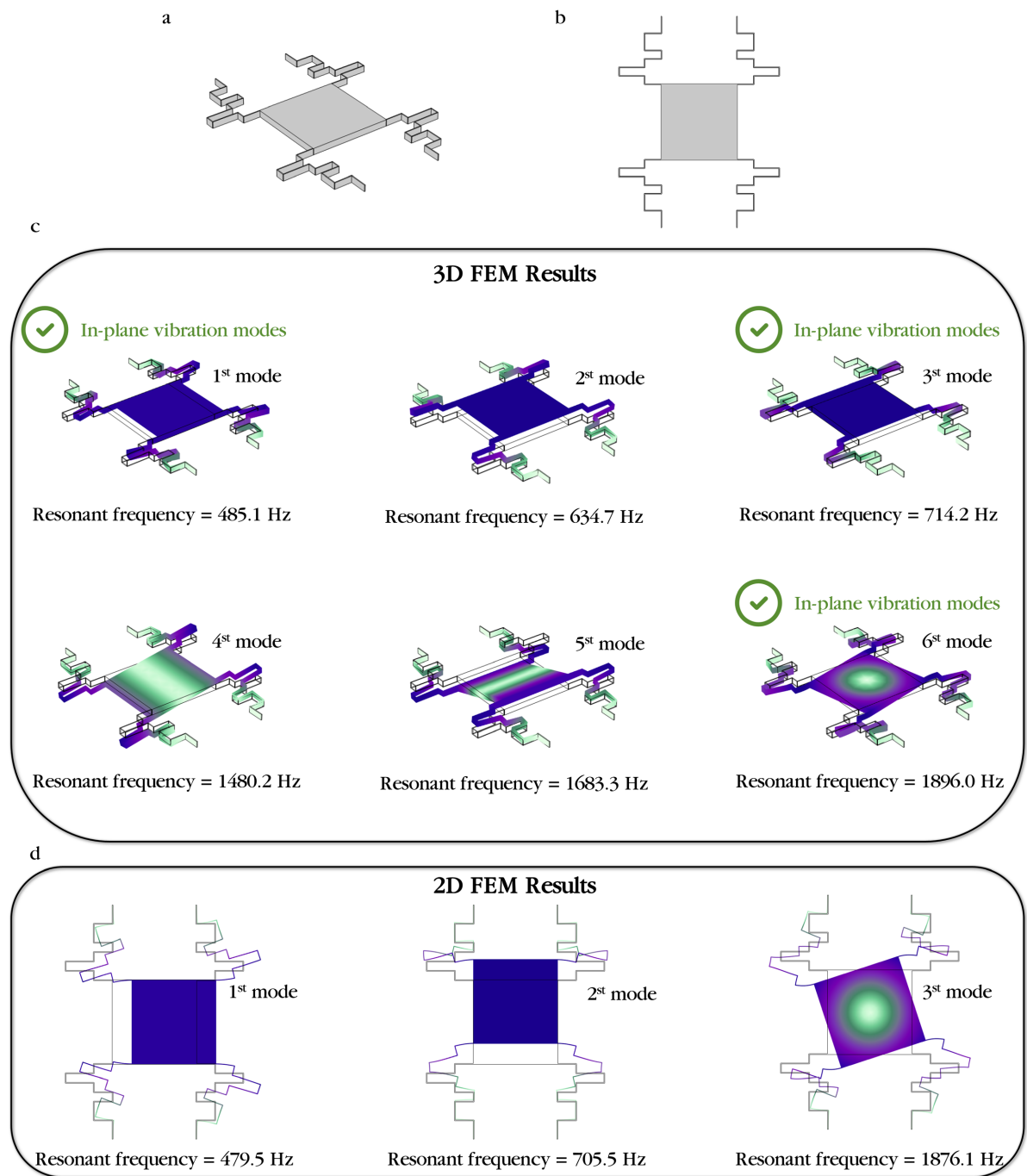


Fig. S17 FEM model and results of the resonator generated with an 8×8 grid. a. 3D FEM model. b. 2D FEM model. c. 3D FEM results. d. 2D FEM results.

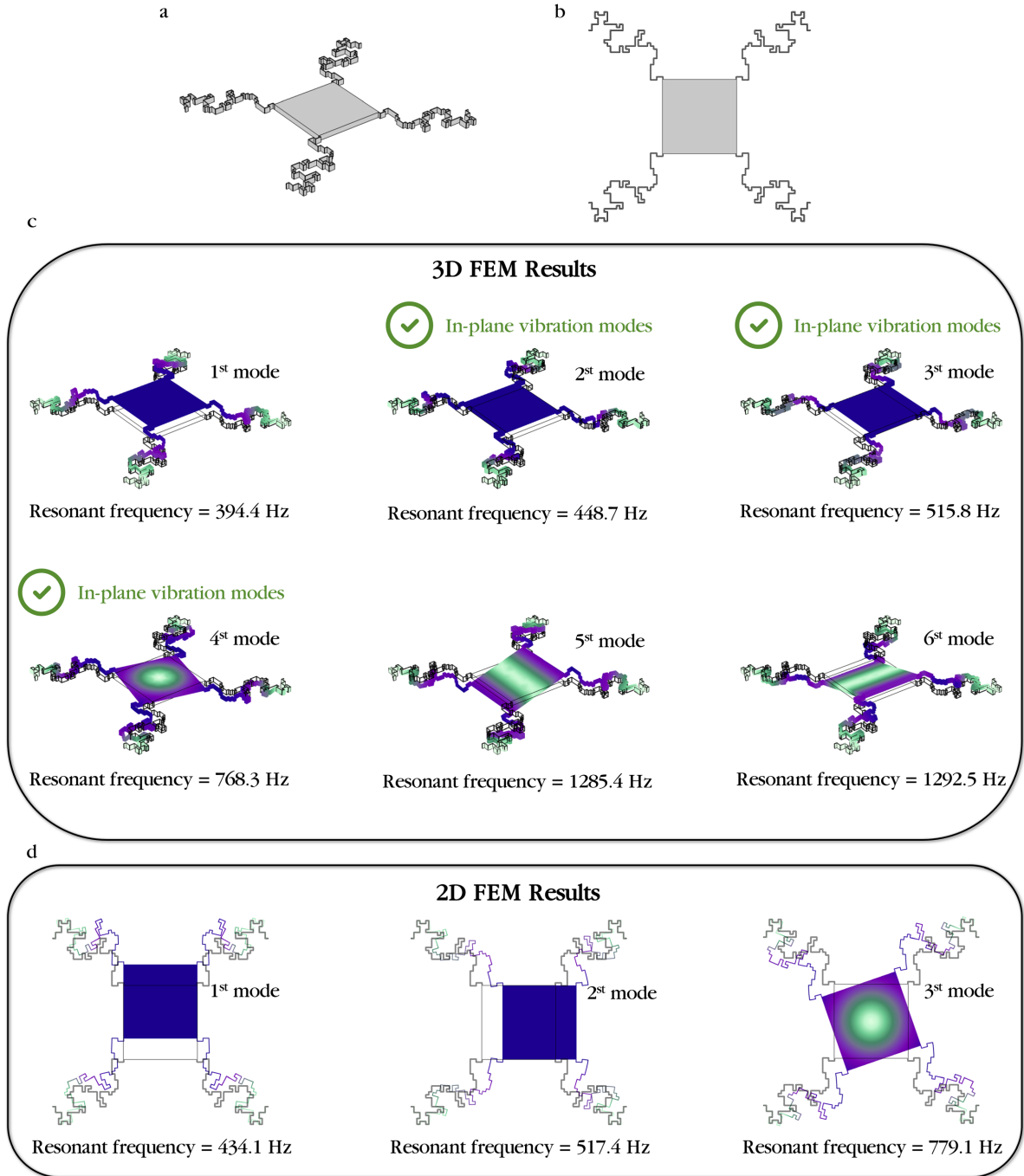


Fig. S18 FEM model and results of the resonator generated with a 24×24 grid. a. 3D FEM model. b. 2D FEM model. c. 3D FEM results. d. 2D FEM results.

As illustrated in Fig. S17 and Fig. S18, three-dimensional (3D) and two-dimensional (2D) FEM analyses were conducted for resonators generated with 8×8 and 24×24 grids, respectively. The 3D FEM models were obtained by extruding the corresponding 2D FEM models, as shown in Fig. S17a and Fig. S17b. The first six vibration modes and resonant frequencies were extracted using 3D FEM, with the 1st, 3rd, and 6th modes corresponding to in-plane vibration modes, as illustrated in Fig. S17c.

For 2D FEM analysis, the z -direction displacement (perpendicular to the resonator plane) was constrained to zero, restricting the resonator motion to the x - y plane, thereby directly obtaining the physical properties of in-plane vibration modes. The first three vibration modes and resonant frequencies obtained directly from 2D FEM (Fig. S17d and Fig. S18d) show excellent agreement with the corresponding in-plane vibration results from 3D FEM. Regarding computational efficiency, the 2D FEM analysis required 8 s per simulation for the 8×8 grid structure compared to 25 s for 3D FEM (68% efficiency improvement), while the 24×24 grid structure required 40 s for 2D FEM versus 180 s for 3D FEM (78% efficiency improvement). In terms of computational accuracy, the deviation between resonant frequencies obtained from 2D and 3D FEM remained consistently below 2% for different grid densities, as listed in Table S13. These results demonstrate that 2D FEM can rapidly characterize the physical properties of in-plane vibration modes while maintaining accuracy with approximately 70% faster analysis speed. Therefore, for this study, 2D FEM serves as an effective alternative to 3D FEM for resonator physical property analysis.

Table S13 Comparison of 3D and 2D FEM Results.

Grid density	Resonant frequency order	3D FEM Results (Hz)	2D FEM Results (Hz)	Result deviation
8×8	First-order	485.1	479.5	1.15%
	Second-order	714.2	705.5	1.22%
	Third-order	1896	1876.1	1.05%
24×24	First-order	448.7	444.8	0.87%
	Second-order	515.8	517.4	-0.31%
	Third-order	768.3	779.1	-1.41%

Supplementary Note 10

Resonator Fabrication and Characterization

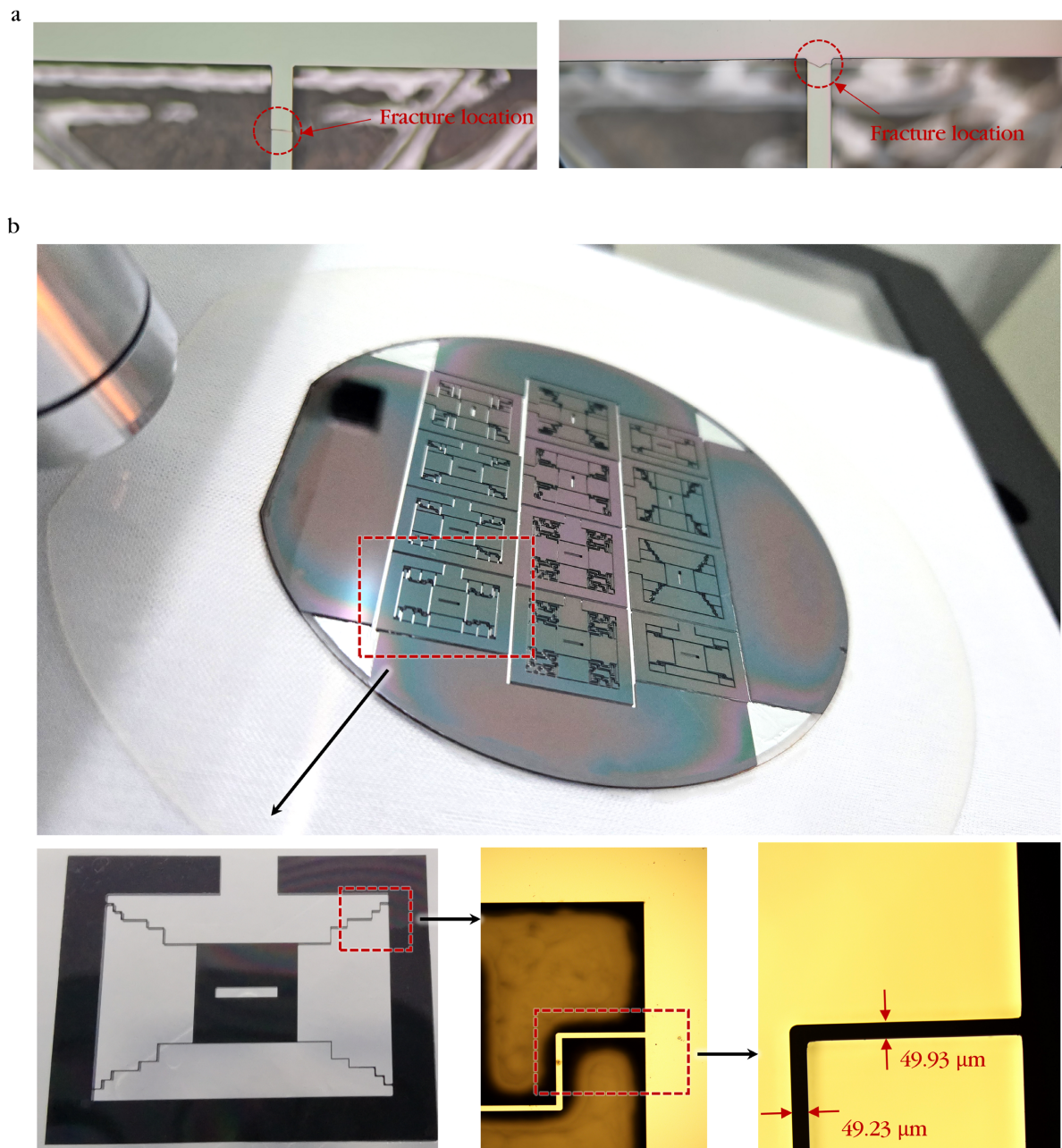


Fig. S19 Fabrication and characterization of the resonators. a. Fracture of the elastic beams in the resonators generated by the high-density grid. b. Fabricated resonators and their local characterization.

The resonator structures generated with high-density grids exhibit finer features, making them prone to fracture during fabrication and handling processes, which significantly reduces the production yield. Fig. S19a demonstrates the fracture of elastic beams in resonators designed with a high-density grid (20×20) during manufacturing. Through extensive process optimization, three grid specifications

were ultimately selected: 8×8 (segment length 500 μm), 10×10 (segment length 400 μm), and 12×12 (segment length 330 μm), achieving a yield exceeding 90%. The resonators were fabricated on 4-inch diameter wafers with a thickness of 400 μm . To ensure machining precision, the etching aspect ratio was strictly maintained below 10:1, leading to a beam width of 50 μm . Each resonator incorporates annular release trenches (20 μm width) and employs stealth dicing technology for device separation. Throughout the design process, all elastic beam junctions feature 5 μm radius fillets, with finite element analysis confirming this approach reduces maximum Von Mises stress by over 30%. Fig. S19b provides detailed images of the fabricated resonators. The final resonator structures demonstrate complete integrity, with key dimensional errors measured via interferometry verified to be within $\pm 0.5 \mu\text{m}$, as shown in Fig. S19b.

Supplementary Note 11

Environmental Adaptability Verification

To evaluate the effect of environmental temperature on the accuracy of ResNES, supplementary temperature-adaptability experiments were conducted. The temperature test points were set at 0°C, 18.2°C, 25°C, 40°C, 60°C, and 80°C.

The temperature control was achieved using an SPX (MJX)-50B thermal chamber, which provides a stable thermal environment with an operational range of 0 °C to 80 °C. The temperature control accuracy is ± 0.1 °C, and under no-load conditions, temperature fluctuations are confined within ± 0.5 °C, providing a consistent thermal baseline for experimental measurements. The experimental procedure is as follows.

First, a piezoelectric ceramic disc was affixed to the underside of the MEMS resonator. After allowing the system to reach thermal equilibrium at each target temperature—with a stabilization period of at least 30 minutes—the resonator was excited into free-decay vibration via the piezoelectric actuator. An LDV was aligned horizontally with the resonator. The LDV laser beam was directed through an optical window on the thermal chamber and focused onto the proof mass of the resonator to capture its vibrational response. The resonant frequencies corresponding to different vibrational modes were subsequently extracted through fast Fourier transform analysis. To ensure statistical reliability, ten complete excitation–acquisition cycles were conducted at each temperature point, and the average value was adopted as the final measured result. This non-contact measurement methodology prevents the introduction of extraneous mechanical loads or thermal perturbations within the chamber, thereby preserving the integrity and authenticity of the experimental data.

As shown in Fig. S20 and Table S14, for the two samples employed in the supplementary experiments, the resonant frequencies of both samples exhibit a systematic upward trend with increasing temperature over the range of 0°C to 80°C. This behavior originates from the packaging configuration of the resonators used in this work, where the device frame is bonded to a glass substrate and a glass cover to minimize environmental interference. When the temperature rises, the glass substrate, which has a higher coefficient of thermal expansion than silicon, undergoes greater thermal expansion. Consequently, a sustained axial tensile stress is applied to the resonator's elastic beams via the anchors, significantly increasing the equivalent stiffness of the resonator. This thermally induced stress effect, introduced by the packaging, completely outweighs the softening effect resulting from the decrease in the Young's modulus of silicon with increasing temperature, thereby becoming the dominant factor

governing the frequency-temperature characteristics. Ultimately, the overall stiffness of the resonator increases with temperature. This mechanism has been extensively observed and validated in various studies on MEMS resonators and sensors [7][8][9][10].

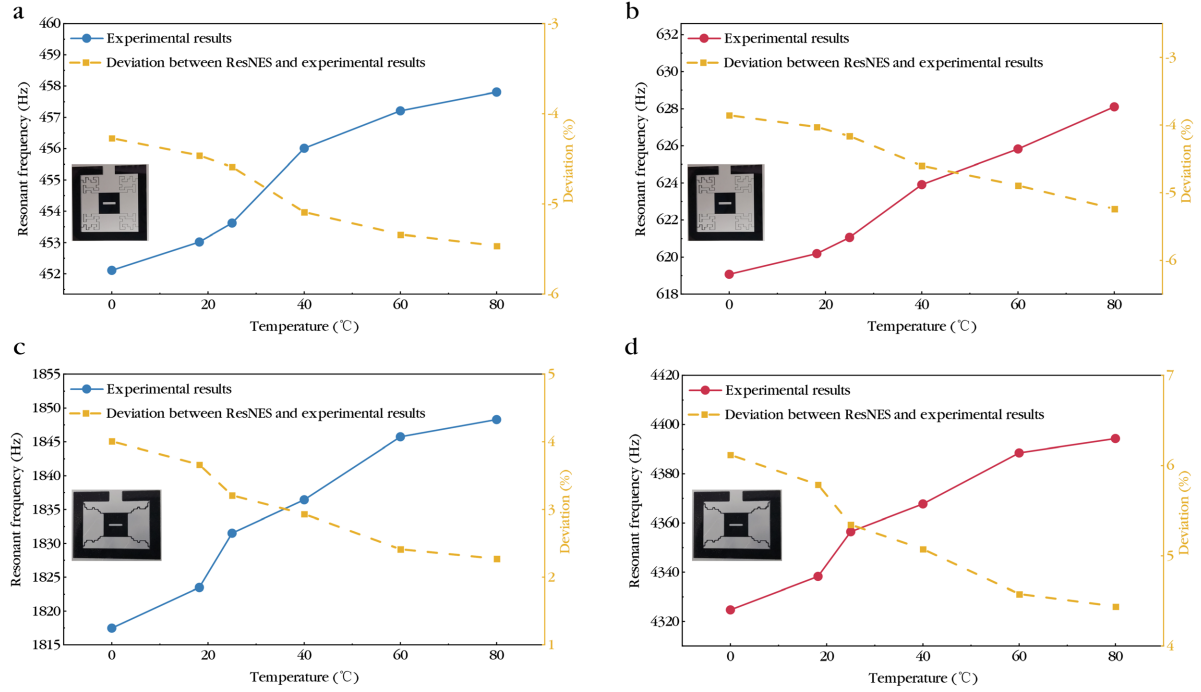


Fig. S20. Measured resonant frequencies and ResNES prediction deviations at different temperatures. a Sample 1, first-order resonant frequency. b Sample 1, second-order resonant frequency. c Sample 6, first-order resonant frequency. d Sample 6, second-order resonant frequency.

Table S14. Experimental results and ResNES prediction deviations.

Sample number	Frequency order	0°C		18.2 °C		25 °C		40°C		60°C		80°C	
		Frequency (Hz)	Deviation	Frequency (Hz)	Deviation	Frequency (Hz)	Deviation	Frequency (Hz)	Deviation	Frequency (Hz)	Deviation	Frequency (Hz)	Deviation
1	First-order	452.11	-4.27%	453.02	-4.46%	453.62	-4.59%	456.01	-5.09%	457.21	-5.34%	457.81	-5.46%
	Second-order	619.07	-3.86%	620.19	-4.03%	621.06	-4.16%	623.91	-4.60%	625.83	-4.89%	628.11	-5.24%
6	First-order	1817.47	4.01%	1823.49	3.66%	1831.51	3.21%	1836.44	2.93%	1845.74	2.41%	1848.29	2.27%
	Second-order	4324.71	6.12%	4338.27	5.79%	4356.49	5.35%	4367.74	5.07%	4388.51	4.58%	4394.31	4.44%

It is noteworthy that although variations in ambient temperature can cause fluctuations in the measurement results, the change in resonant frequency induced by an 80°C temperature variation

remains typically below 1.5%, and the resulting variation in ResNES prediction deviation is well constrained within 1.8%. Therefore, we consider the fluctuation in ResNES prediction accuracy due to extreme environmental temperatures to be within an acceptable range.

Supplementary Note 12

Construction and performance comparison of the Adaptive Elite Learning Particle Swarm Optimization (AELPSO) Algorithm

Intelligent optimization algorithms, such as particle swarm optimization (PSO), play a crucial role in solving multimodal optimization problems. These algorithms are simple, easy to implement, and do not require mathematical operations such as differentiation. They are widely applicable, converge rapidly, and are computationally efficient. However, they are susceptible to converging to local optima. To enhance the algorithm's performance for the optimization problem in this study, we have leveraged its strengths while mitigating its weaknesses.

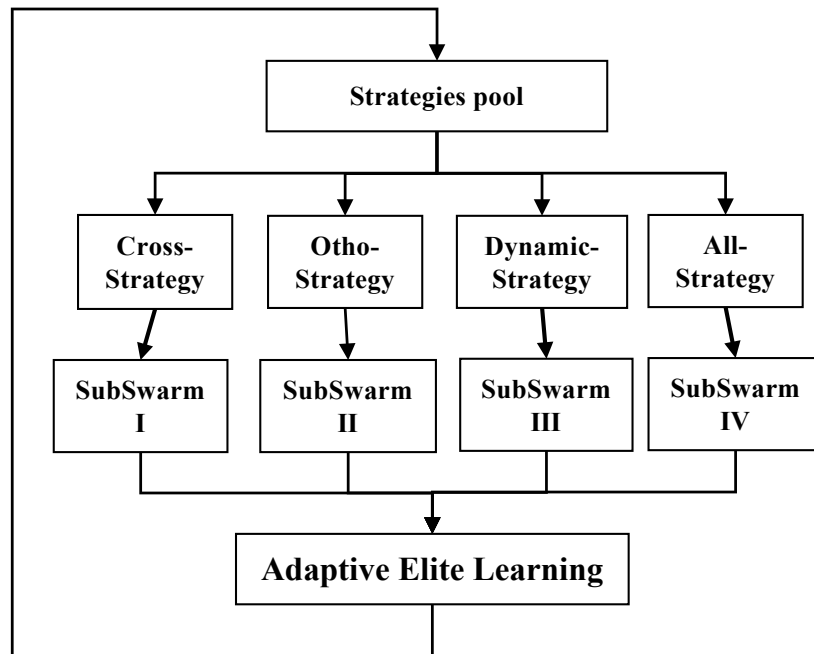


Fig. S21 Framework of AELPSO.

The proposed optimization algorithm, Adaptive Elite Learning Particle Swarm Optimization (AELPSO) ^[11], integrates concepts from dynamical systems theory, population diversity analysis, and computational complexity analysis to guide the selection of sub-strategies and adaptively partition the population. AELPSO incorporates four distinct sub-strategies: Dynamic Strategy, Cross Strategy, All Strategy, and Otho Strategy, collectively forming the Collaborative Optimization with Diverse Algorithms (CODA) framework. The core innovation of AELPSO lies in its multi-strategy cooperative mechanism, enabling the co-evolution of four subswarms, dynamically balancing global exploration and local exploitation. Furthermore, the algorithm employs an Adaptive Elite Learning (AEL) mechanism, wherein elite particles are adaptively assigned to execute a broader range of strategies, thus

enhancing the diversity of search behaviors.

AELPSO consists of two primary modules: a multi-population framework and an adaptive elite learning mechanism. The schematic flowcharts of the algorithm are shown in Fig. S21 and Fig. S22.

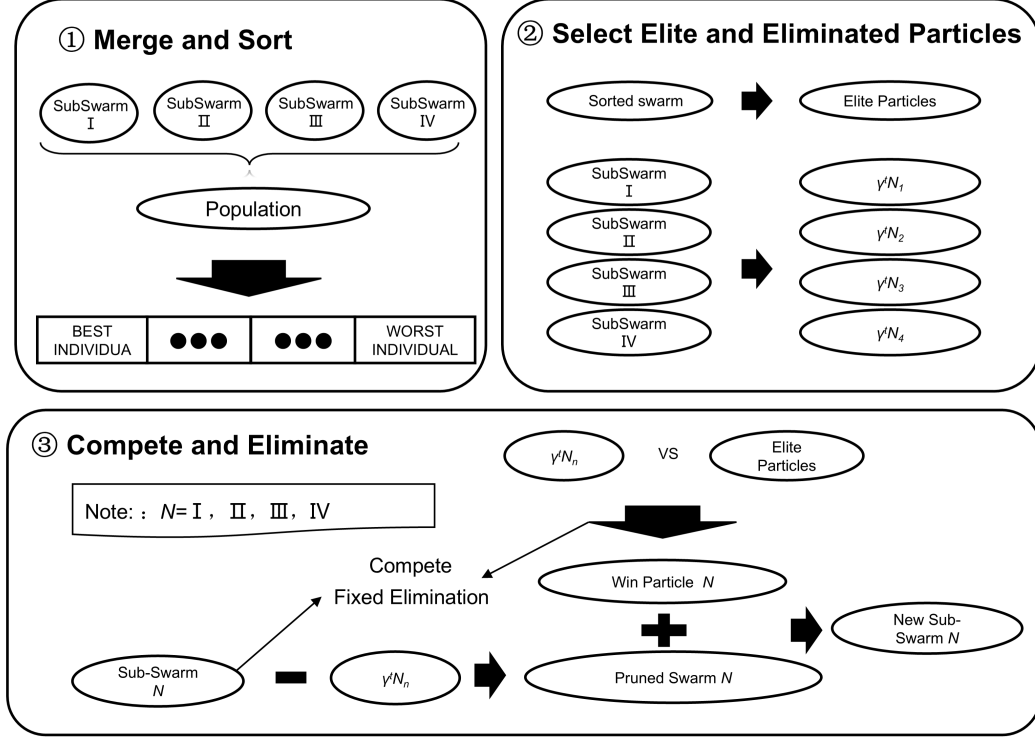


Fig. S22 Flowchart of AEL.

(1) Strategy Pool Construction

Strategy ‘C’^[12]: Two particles are randomly selected from the population. The particle with the better personal best fitness value is identified, and a crossover operation is performed between its personal best and the personal best of the particle currently updating its velocity. This process constructs the learning target $p_{(1)}$, and the velocity update formula is defined as:

$$v_{i,j}^{T+1} = \omega_1 * v_{i,j}^T + c * r_{i,j} * (p_{(1) i,j} - x_{i,j}^T) \quad (3)$$

Strategy ‘O’^[13]: The method used in Strategy 'A' is applied to create small regions within the population, such that the optimal particle in each region is orthogonally combined with the particle updating its velocity. This forms the learning objective $p_{(2)}$, with the corresponding velocity update formula given by:

$$v_{i,j}^{T+1} = \omega_2 * v_{i,j}^T + c * r_{i,j} * (p_{(2) i,j} - x_{i,j}^T) \quad (4)$$

Strategy ‘D’^[14]: The classical PSO speed update strategy is employed, which calculates the Euclidean distance between particles to assess the population’s state. Based on this state, different

values are assigned to the inertia weight ω and acceleration coefficient c . The learning objective $p_{(3)}$ is then constructed, and particles in the convergence state undergo mutation to check if they are trapped in local optima. The velocity update formula for this strategy is:

$$v_{i,j}^{T+1} = \omega_3 * v_{i,j}^T + c_1 * r_{i,j} * (p_{i,j} - x_{i,j}^T) + c_2 * r_{i,j} * (p_{(3)} - x_{i,j}^T) \quad (5)$$

Strategy ‘A’^[15]: The ring topology is used to adjust the positional relationships between particles. After the adjustment, each particle is surrounded by two others, forming a small region. The average of the individual history optima of these three particles is then used to construct the learning objective $p_{(4)}$, and the velocity update formula is:

$$v_{i,j}^{T+1} = \omega_4 * v_{i,j}^T + c * r_{i,j} * (p_{(4)i,j} - x_{i,j}^T) \quad (6)$$

The j -th dimensional position update formula for particle i of the above strategy:

$$x_{i,j}^{T+1} = x_{i,j}^T + v_{i,j}^{T+1} \quad (7)$$

(2) Adaptive Elite Learning

Merge and Sort

After executing policies C, O, D, and A across the four sub-populations, the particles are fused into a single larger population. This new population is then sorted uniformly based on the fitness values $f(x)$, generating sequences of ordered particles and recording their origins from the respective sub-populations.

Select Elite and Eliminated Particles

Let N_1, N_2, N_3 , and N_4 denote the sizes of the four sub-populations, and then select $\gamma^t N_n$ and the first $\gamma^t N_n$ ($n = 1, 2, 3, 4$) particles from each sub-population and the sorted large population, respectively. These particles are compared based on their individual optimal values. The first $\gamma^t N_n$ selected from the large population are designated as elite particles, with the selection ratio defined as:

$$\gamma^t = \phi_{\min} + (\phi_{\max} - \phi_{\min}) \times \left(\frac{t}{T} \right)^2 \quad (8)$$

For each execution of the adaptive elite learning method, $\gamma^t N_n$ ($n = 1, 2, 3, 4$) particles are selected from each sub-population and marked as eliminated particles.

Compete and Eliminate

If the individual optimal value of a particle selected from each sub-population is worse than that of a particle from the larger population, the corresponding information (including position, velocity, and individual optimal experience) of the particle in the sub-population is replaced by the information

of the superior particle from the larger population. In other words, the elite particle replaces the eliminated particle.

In summary, the adaptive elite learning mechanism described here allows higher-performing particles to surpass single search patterns. The algorithm is evaluated using the CEC2020 and CEC2022 benchmarks, and its performance is compared with that of other algorithms. The results, shown in Fig. S23, demonstrate that the AELPSO algorithm significantly outperforms the other algorithms in both test sets.

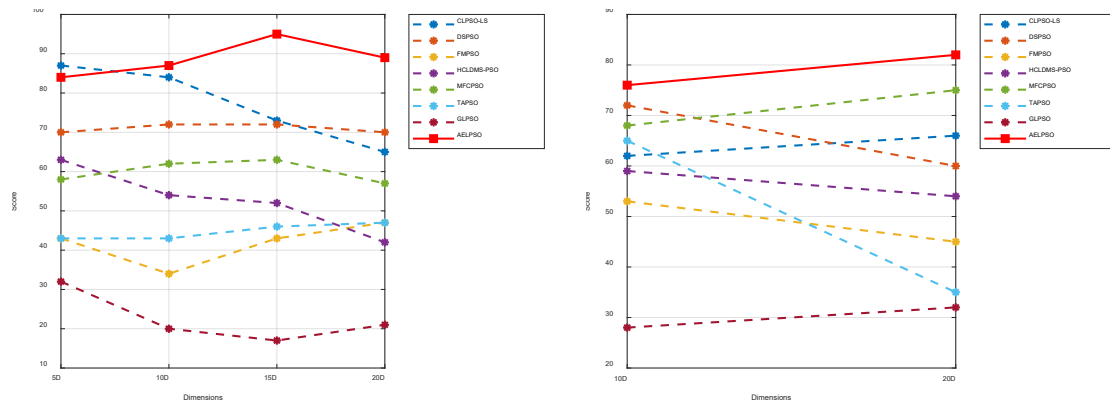


Fig. S23 Comparison of optimization results for CEC2020 and CEC2022

Supplementary References

1. Chen, Chen, et al. Machine learning-driven discovery of high-performance MEMS disk resonator gyroscope structural topologies. *Microsystems & Nanoengineering*. 10, 161 (2024).
2. Cheng, Xiong, et al. Generating multiple distinct feasible solutions for MEMS accelerometers using deep learning. *IEEE Sensors Journal* (2024).
3. Guo, Ruiqi, et al. Deep learning for non-parameterized MEMS structural design. *Microsystems & nanoengineering*. 8, 91 (2022).
4. Chimeh, Hamidreza Ehsani, et al. Deep-learning-based optimization for a low-frequency piezoelectric MEMS energy harvester. *IEEE Sensors Journal*. 21, 21330-21341 (2021).
5. Huang, Gao, et al. Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2017).
6. Huang, Xin, et al. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678* (2020).
7. Xiaorui, Bie, et al. Analysis of Thermal Hysteresis in Silicon Resonant Accelerometers Caused by Packaging Stress. 2025 26th International Conference on Electronic Packaging Technology (ICEPT). IEEE. (2025).
8. Jiang, B., et al. Analysis of frequency drift of silicon MEMS resonator with temperature. *Micromachines*. 12, 59 (2021).
9. Cao, Yunqi, and Nelson Sepúlveda. Interface stress for bidirectional frequency tuning of prebuckled vanadium dioxide MEMS resonators. *Advanced Materials Interfaces*. 6, 1900887 (2019).
10. Li, Zheng, et al. Thermal Stress Minimization for Torsional Pendulum MEMS Gyroscopes Via Center-Supported Structural Design. *IEEE Sensors Journal* (2025).
11. Wu, Shuang, et al. A μ Gal MOEMS gravimeter designed with free-form anti-springs. *Nature Communications*. 16, 1786 (2025).
12. Liang, Jing J., et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE transactions on evolutionary computation*. 10, 281-295 (2006).
13. Zhan, Zhi-Hui, Jun Zhang, and Ou Liu. Orthogonal learning particle swarm optimization. *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. (2009).

14. Zhan, Zhi-Hui, et al. Adaptive particle swarm optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 39, 1362-1381 (2009).
15. Mendes, Rui, James Kennedy, and José Neves. The fully informed particle swarm: simpler, maybe better. IEEE transactions on evolutionary computation. 8, 204-210 (2004).