Supplementary Information for Deep Learning-Empowered Programmable Topological Circuits

Hao Jia,[1,*] Shanglin Yang,[2,*] Jiajun He,[1] Shuo Liu,[3] Haoxiang Chen,[4] Ce Shang,[5,†] Shaojie Ma,[6] Peng Han,[7] Ching Hua Lee,[8] Zhen Gao,[9] Yun Lai,[10,‡] and Tie Jun Cui[3,§]

[1]*School of Physical Science and Technology, Lanzhou University, Lanzhou 730000, China.*

[2]*School of Optoelectronic Engineering, Xidian University, Xi'an 710071, China*

[3]*State Key Laboratory of Millimeter Waves, Southeast University, Nanjing 210096, China.*

[4]*International School of Microelectronics, Dongguan University of Technology, Dongguan 523808, Guangdong, China.*

[5]*Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China*

[6]*Shanghai Engineering Research Centre of Ultra Precision Optical Manufacturing, Department of Optical Science and Engineering, School of Information Science and Technology, Fudan University, Shanghai 200433, China*

[7]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*

[8]*Department of Physics, National University of Singapore, Singapore 117551, Republic of Singapore*

[9]*Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China.*

[10]*National Laboratory of Solid State Microstructures, School of Physics, and Collaborative Innovation Center of Advanced Microstructures, Nanjing University, Nanjing 210093, China*

(Dated: August 27, 2025)

* These authors contributed equally

CONTENTS

† shangce@aircas.ac.cn
‡ laiyun@nju.edu.cn
§ tjcui@seu.edu.cn

S1. PROGRAMMABLE TOPOLECTRICAL CIRCUIT PLATFORM CONSTRUCTION

A. Mapping the Hamiltonian to circuit Laplacian

The key point of a topolectrical circuit is that one can fully map the tight-binding Hamiltonian in condensed matter physics to circuit Laplacians (admittance matrices) [1]. The large variety of electrical elements and variable connection modes allows the researchers to realize a wide variety of topological states, some of which are extremely challenging to observe in condensed matter systems. In particular, the off-site hoppings and on-site potentials host generous freedoms in circuits, such as the strength, direction, and dimension, which enable us to introduce interactions between two arbitrary nodes and control the on-site energy for each node.

According to Kirchhoff's laws, the response of an electrical circuit can be described by the equation of motion:

$$\frac{d}{dt}I_a = C_{ab}\frac{d^2}{dt^2}V_b + L_{ab}V_b, \tag{S1}$$

where I_a is the current flowing out of node a and V_b is the electrical potential at node b. C_{ab} and L_{ab} are the capacitance and conductance between nodes a and b, respectively. When we apply an alternating voltage $V(t) = V(0)e^{i\omega t}$ to the circuit, Eq. (S1) yields:

$$I_a = \sum_b \left(i\omega C_{ab} + \frac{1}{i\omega}L_{ab}\right)V_b = \sum_b J_{ab}(\omega)V_b, \tag{S2}$$

with the matrix form parameter:

$$\boldsymbol{J}(\omega) = i\omega\boldsymbol{C} + \frac{1}{i\omega}\boldsymbol{L}, \tag{S3}$$

where $\boldsymbol{J}(\omega)$ is defined as the circuit Laplacian and ω is the frequency. $\boldsymbol{C}$ and $\boldsymbol{L}$ are the Laplacian matrices of the capacitance and inverse inductance, respectively. The diagonal and off-diagonal elements represent self-admittance via a certain node and mutual admittance between two nodes. We use ω for steady-state analysis of the circuit and obtain an adiabatic continuum of spectra $j(\omega)$ corresponding to $\boldsymbol{J}(\omega)$. As the capacitance and inductance explicitly depend on it, the frequency ω of the driving voltage is a central tuning parameter of topolectrical circuits. In the programmable circuit design, the off-site hopping terms are proportionally mapped to the off-diagonal element of a Hamiltonian $H(i, j)$ by

linking capacitance $C_{E_{\mathbf{r},\mathbf{r}'}} = \kappa \times H(i,j)$, and the on-site potentials V represented by diagonal elements of Hamiltonian are also proportionally mapped to the circuit by tailoring the on-site capacitance so that the condition $C_{V_\mathbf{r}} = \kappa \times H(i,i)$ is fulfilled. A special example is the case at the central frequency ω_c, where the parallel resonance results in the on-site potential equaling zero, and the diagonal terms in the circuit Laplacian and Hamiltonian are all zero. The relationship between J and H is:

$$J = i\omega\kappa H. \tag{S4}$$

In this way, the topolectrical circuit can fully reflect the topological characteristics that a certain Hamiltonian decides.

B. Programmable on-site and off-site terms design

Based on the mapping from Hamiltonian to circuit Laplacian, engineering the on-site potentials and off-site hopping terms can contribute to tuning the capacitance or inductance of the elements at the vortex and edge (connection) positions of the circuit. Normally, tunable capacitance is more practical to realize than inductance. In our circuit design, the varactor diodes are used to provide variable capacitance. The detailed circuit units for off-site and on-site capacitance are shown in Fig. **S1**.

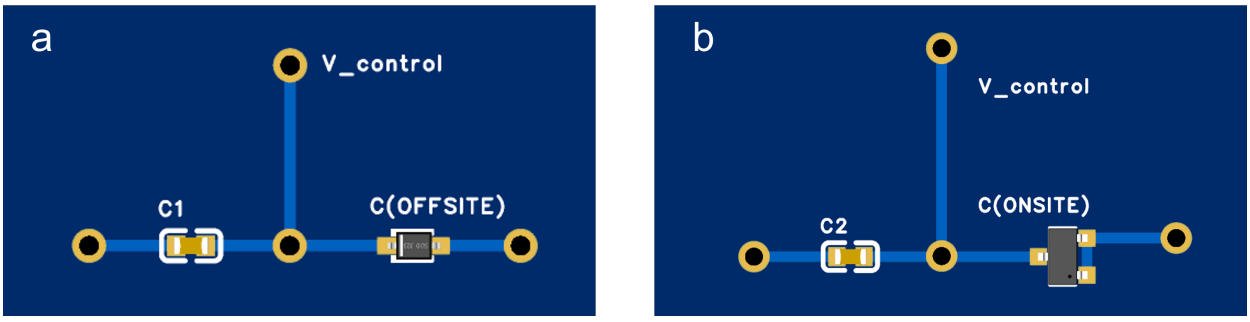

FIG. **S1**. **Circuit configuration of variable capacitance.** (a) Off-site capacitance unit. (b) On-site capacitance unit.

For off-site capacitance, varactor diode (SKYWORKS, SMV 1702-011LF) is in serial with a 1.5 nF $\pm$ 5% ceramic capacitor C_1 (MuRata, GCM1885C2A152JA16D). The serial ceramic capacitor is used to separate the bias voltage signal from other unit cells and tune the fluctuation range of capacitance. For on-site capacitance, a Hyperabrupt Junction Tuning

Varactor diode (SKYWORKS, SMV 1470-004LF) with two diodes back-to-back connected in the package is externally shunted on PCB, then in series with the 10 $\mu F \pm 10\%$ ceramic capacitor C_2 (MuRata, GRM21BR61H106KE43L). Control voltage $V_{control}$ is swept by a high precision programmable digital synthetic power supply from $0V$ to $10V$. The capacitance-voltage calibration curve is shown in Fig. **S2**. The tunable range is [10 pF, 100 pF] for off-site capacitance, and [18 pF, 240 pF] for on-site capacitance.

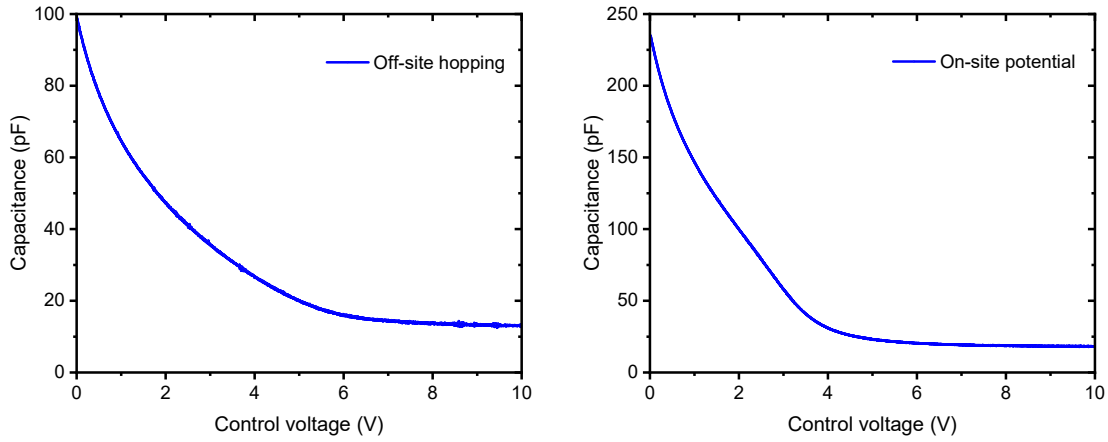

FIG. **S2**. **Capacitance calibration curves for off-site and on-site terms.** (a) Off-site hopping term. (b) on-site term.

C. The realization of deep learning empowered programmable topolectrical circuits

In this work, the fundamental framework of the whole programmable topolectrical circuit is a honeycomb-like circuit lattice with a triangular geometry. Our prototype has a scale $N = 9$, which includes 100 sites distributed in triangular geometry, and 135 edge connections (hoppings), i.e., 100 on-site terms and 135 off-site terms in the Laplacian.

A four-layer PCB board with topological connections, signal +, signal -, and a grounded shielding layer is designed. An in-house 256-channel programmable digital synthetic power supply is connected to each varactor diode to provide the control voltage. The minimum voltage step can reach 0.15 mV. An in-house library is constituted to build the bridge between hardware driving and the interface of the GUI and deep-learning framework, so that the hardware and software can cooperate closely. Our board is shown in Fig. **S3**. The

bird's view of the whole system is shown in Fig. S4.

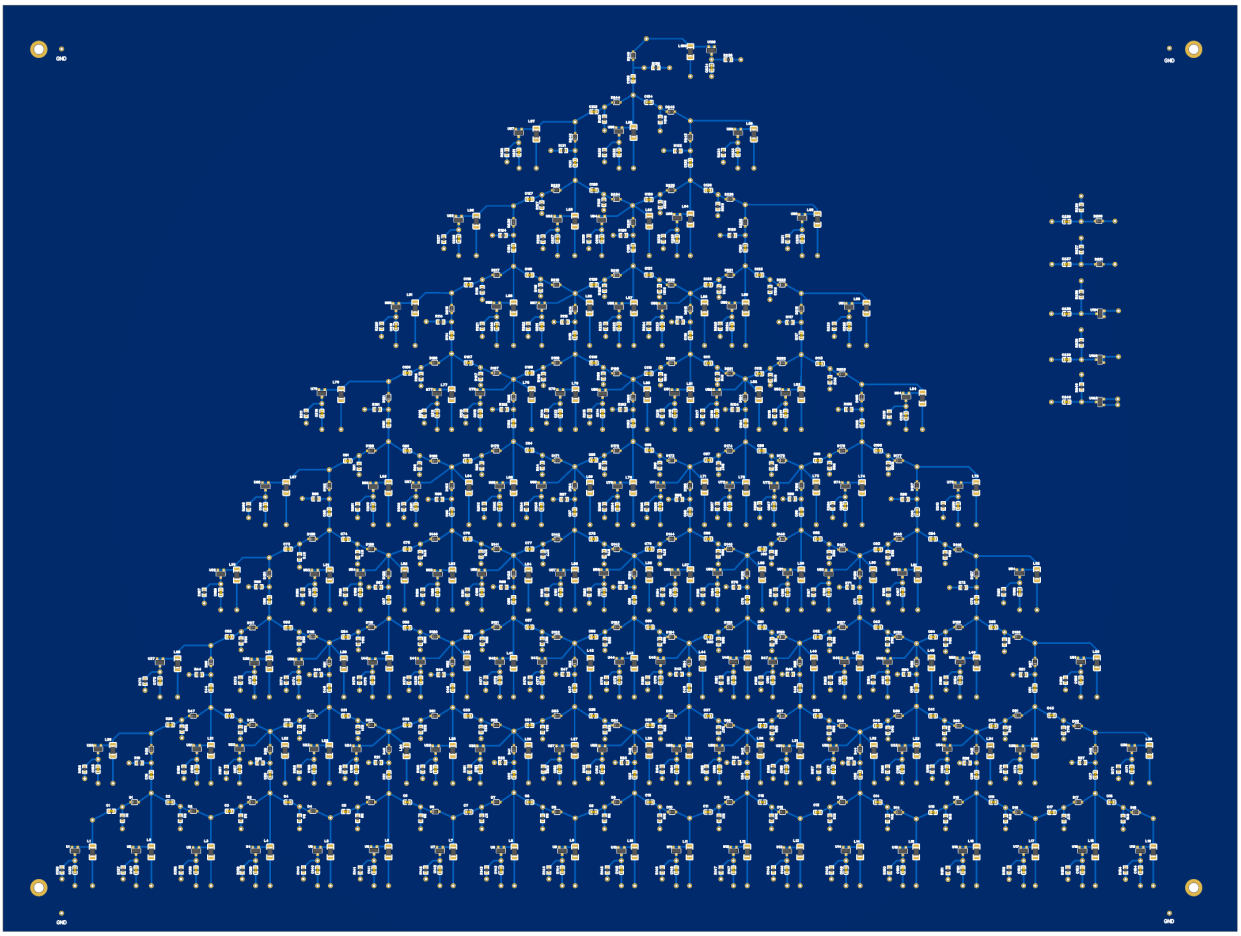

FIG. **S3**. **The designed circuit board.**

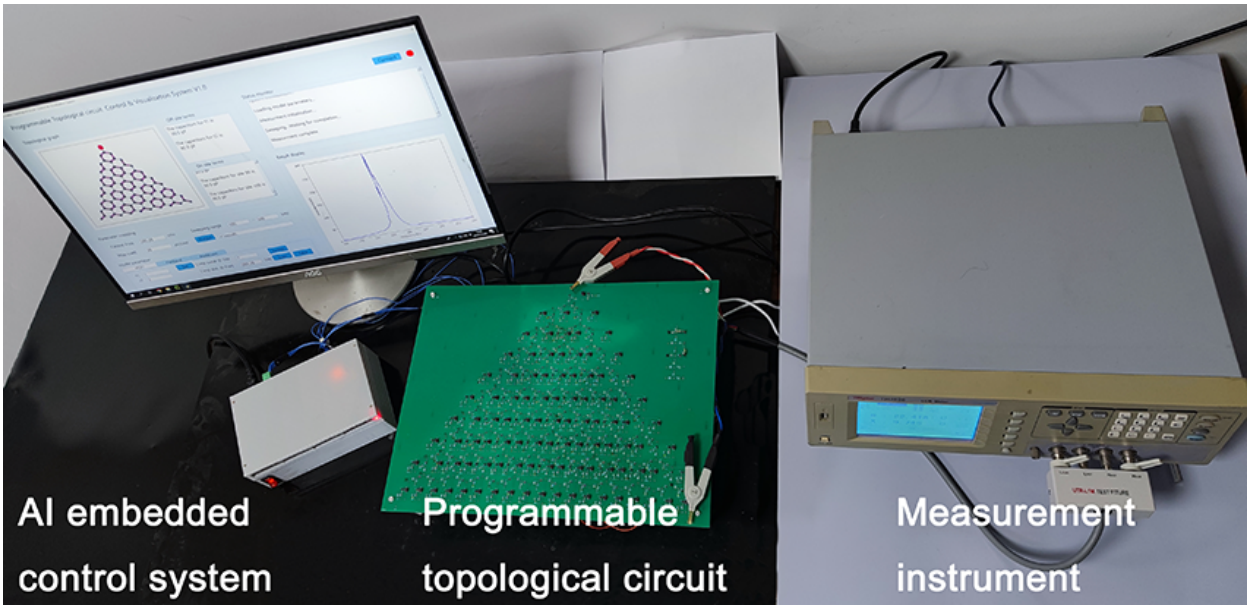

FIG. **S4. The bird's view of the whole system configuration.**

D. Extension of off-site hopping unit to realize negative and complex values

Enabling all forms of couplings—allowing the hopping to be tuned through both positive and negative values, and even to complex values—would provide the most complete universality of the platform. However, constrained by current electronic components and practical engineering considerations, realizing completely arbitrary interaction coupling forms is still very challenging. In the following, we give some extension variable-unit designs to achieve more general couplings, including the realization of negative and complex values.

1) Realization of positive and negative-valued hopping

To realize positive–negative valued hopping within a passive-device regime, we designed a unit, as shown in Fig. **S5**a: between two nodes A and B, we connect in parallel the variable capacitor used in the main text and an inductor. Since an inductor is equivalent to a negative capacitance in the effective low-frequency description, one can choose appropriate values such that the effective hopping can be tuned across positive and negative values.

The operating principle of the module can be derived as follows. Within the framework of topolectrical circuit theory (Supplementary information S1.A), the Hamiltonian H of the target lattice and the circuit Laplacian $J(\omega)$ are related by:

$$H \propto \mathrm{i}J(\omega), \tag{S5}$$

where, J is

$$\begin{aligned} J(\omega) &= N_{ab}(\omega) + \delta_{ab} W_a(\omega), \\ N_{ab}(\omega) &= -\mathrm{i}\omega C_{ab} + \frac{\mathrm{i}}{\omega L_{ab}} \end{aligned} \tag{S6}$$

thus,

$$H_{ab} \propto \omega \left(C_{ab} - \frac{1}{\omega^2 L_{ab}} \right). \tag{S7}$$

Here, N_{ab} and W_{ab} denote the off-diagonal and diagonal terms of the admittance (Laplacian) matrix, respectively. H_{ab} represents the corresponding hopping term in the Hamiltonian. By tuning the capacitance C_{ab}, the value of H_{ab} can be continuously varied, sweeping from positive, through zero, to negative values. We use Murata LXRW0YV201 as a variable capacitance element to provide the separate capacitance and bias voltage. Using the impedance analyzer, we implement different control voltages on such a hopping unit and

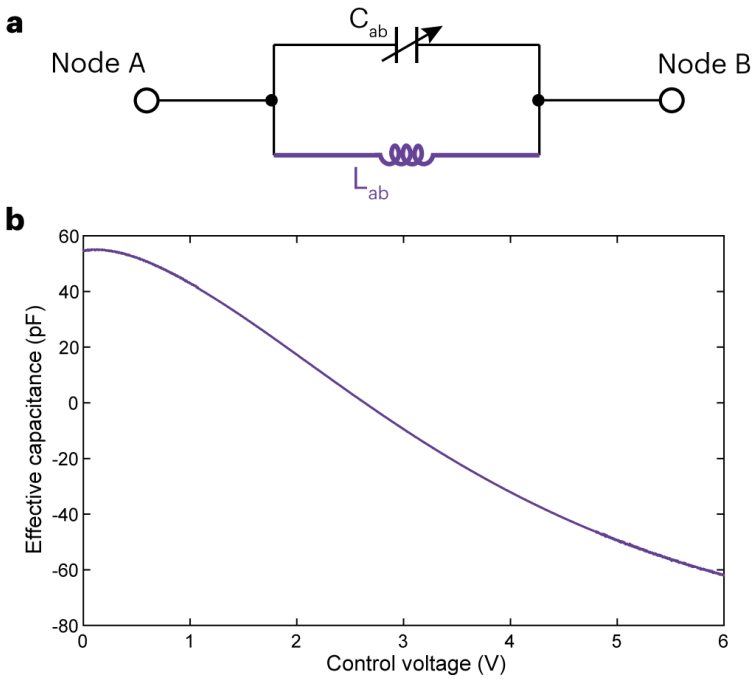

FIG. **S5**. **Hopping unit for positive and negative value** (a) The circuit design of hopping unit. (b) The measured effective capacitance with the tuning control voltage at the central frequency.

measure the capacitance. The experiment measured the effective capacitance with the tuning control voltage shown in Fig. **S5**b, which clearly shows the tuning range from positive across zero to the negative region.

As a proof of concept, we supplement a generic programmable topolectrical circuit built from this unit. As shown in Fig. **S6**, the circuit comprises three nodes A, B, and C. Each pair of nodes is connected by the programmable hopping unit, and each node is connected to ground through an inductance and a compensation capacitance.

To further demonstrate its programmability, we implement custom-defined function curves to different hopping units t_{ab}, t_{ac}, and t_{bc}, which connect the nodes (A, B), (A, C), and (B, C). Without loss of generality, we implement the function as:

$$\begin{aligned} t_{ab}(m,n) &= p_1 \cos\left(\frac{\sqrt{3}}{2}n\right) + p_2 \cos\left(\frac{3}{2}m\right), \\ t_{ac}(m,n) &= p_1 \cos\left(-\frac{3}{4}m + \frac{\sqrt{3}}{4}n\right) + p_2 \cos\left(\frac{3}{4}m + \frac{3\sqrt{3}}{4}n\right), \\ t_{bc}(m,n) &= p_1 \cos\left(\frac{3}{4}m + \frac{\sqrt{3}}{4}n\right) + p_2 \cos\left(-\frac{3}{4}m + \frac{3\sqrt{3}}{4}n\right), \end{aligned} \tag{S8}$$

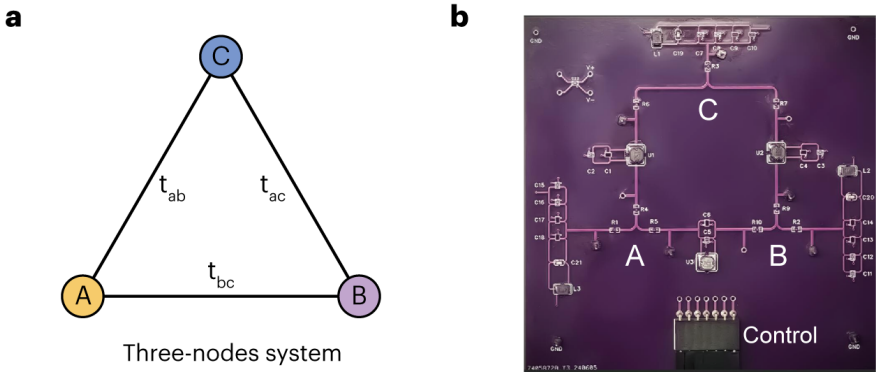

FIG. **S6**. **A circuit demo with positive and negative tunable hopping value.** (a) The schematic of the system with three nodes labeled by A, B and C, t_{ab}, t_{ac}, and t_{bc} represent the hopping value connecting the nodes (A, B), (A, C), and (B, C). (b) The picture of the fabricated PCB with complete circuit elements.

where p_1 ,p_2 are tunable control parameters for system evolution and m ,n are independent variable for the functions. Considering the circuit is reciprocal, the Hamiltonian of such a three-point system can be written as:

$$H(m,n) = \begin{pmatrix} 0 & t_{ab} & t_{ac} \\ t_{ab} & 0 & t_{bc} \\ t_{ac} & t_{bc} & 0 \end{pmatrix}. \tag{S9}$$

As the demonstration, we arbitrarily select the system parameters (p_1, p_2) = (-0.28, -0.72). We fix the value of m or n and sweep the other variable from $-\pi$ to π. Without loss of generality, we arbitrarily select $n = \pi/4$ or $m = -\pi/2$. We continuously load each hopping configuration onto the circuit and experimentally measure the circuit Green's function, so as to get the Laplacian. To demonstrate the accuracy of the programmable hopping term, we show the comparison between the theoretical value and the measured value: the hopping values of t_{ab}, t_{ac}, and t_{bc}, and the eigen-spectrum of such a three-point system. The results are shown in Fig. **S7** below. It can be observed that for arbitrarily selected circuit parameters, the three hopping terms on each side, and the related eigen-spectrum are very precisely alignment between the designed result and measured result. And the hopping values are contentiously varied across positive and negative values.

By changing another system parameters (p_1, p_2) = (-0.5, -0.5). We also fix the value of m or n and sweep the other variable from $-\pi$ to π. We select $n = -\pi$ or $m = \pi/3$, and

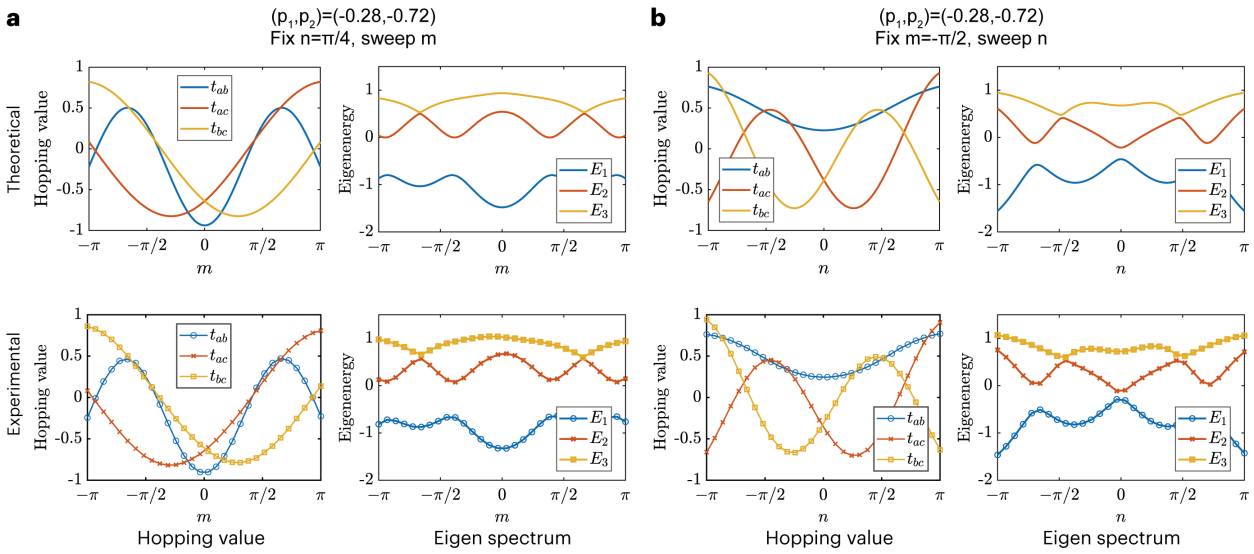

FIG. **S7**. **The comparison between theoretical and experimentally measured hopping values eigen-spectrum for (p_1, p_2) = (-0.28, -0.72).** (a) When variable n is fixed at $\pi/4$ and sweeping m from $-\pi$ to π. (b) When variable m is fixed at $-\pi/2$ and sweeping n from $-\pi$ to π. Points in the experimental curves are measured data.

sweeping the variable n from $-\pi$ to π. The result is shown in Fig. **S8**. It can also show high consistency. In summary, it fully shows the programmable capability of hopping terms.

2) Realization of complex-valued hopping

Going further, we are trying to realize arbitrary complex-valued hoppings for a controlled phase and preserve Hermiticity (i.e., $H_{ab} = H_{ba}^*$). In the current topolectrical circuit system, according to the mapping in Eq. S5, capacitance or inductance contributes to the real part of hopping. As a result, the imaginary part of the hopping requires resistive elements.

Add an effective resistor R_{ab} between node A and node B, the off-site term of the admittance matrix $J(\omega)$ in Eq. S6 becomes

$$N_{ab}(\omega) = -\mathrm{i}\omega C_{ab} - \frac{1}{R_{ab}} \quad . \tag{S10}$$

Based on the previously discussed correspondence between the Laplacian and the Hamiltonian, we can accordingly establish a proportional mapping relationship between the hopping-related parameters. For convenience, we define the complex hopping term $H_{ab} = u + iv$, where u and v are the real and imaginary parts. Considering the hopping term, we have:

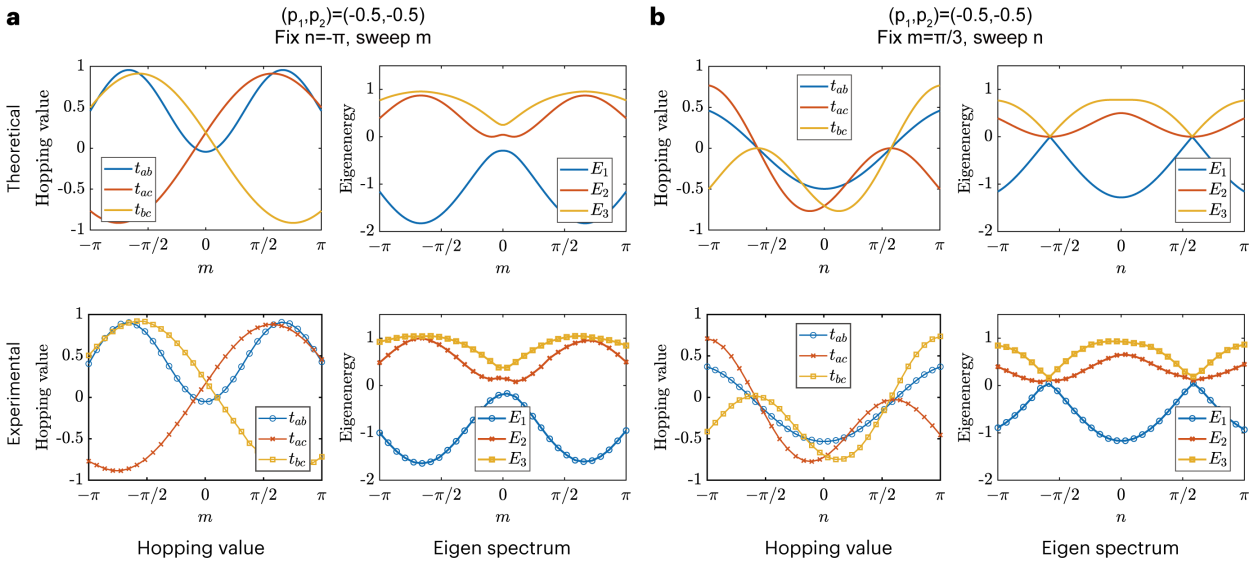

FIG. **S8**. **The comparison between theory and experimentally measured hopping values eigen-spectrum for (p_1, p_2) = (-0.5, -0.5).** (a) When variable n is fixed at $-\pi$ and sweeping m from $-\pi$ to π. (b) When variable m is fixed at $\pi/3$ and sweeping n from $-\pi$ to π. Points in the experimental curves are measured data.

$$\begin{aligned} \mathrm{i}N_{ab} &\leftrightarrow H_{ab}, \\ \mathrm{i}\left(-\mathrm{i}\omega C_{ab} - \frac{1}{R_{ab}}\right) &\leftrightarrow u + \mathrm{i}v, \\ -\frac{1}{R_{ab}} &\leftrightarrow v. \end{aligned} \tag{S11}$$

therefore, to realize complex hopping while maintaining the hermitian of Hamiltonian $H^\dagger = H$ requires $u + \mathrm{i}v$ and $u - \mathrm{i}v$ for H_{ab} and H_{ba}, namely, the resistance R_{ab} should fulfill non-reciprocal condition $R_{ab} = -R_{ba}$.

Passive reciprocal L/C/R networks cannot provide such antisymmetric coupling phases; active components like operational amplifier or analog multiplexer (or magneto-optic components) must therefore be introduced, which greatly increases the difficulty of the experimental implementation and increases the cost of the platform.

We present here a design for general complex hopping. As shown in Fig. **S9**, without loss of generality, we consider the coupling between two nodes A and B. The same as that already shown in the main manuscript, a reciprocal variable capacitor is connected between A and B to supply the real part of the hopping. In parallel, we construct a negative-impedance

converter (NIC) using an operational amplifier and connect a resistor R_{ab} between A and B, which provides non-reciprocal resistance.

As shown in the lower part of Fig.**S9**, considering the virtual-short and virtual-open characteristics of an ideal operational amplifier, we have

$$I_{in} = \frac{(V_a - V_{oa})}{R_1}, \quad I_{out} = \frac{(V_a - V_b)}{R}, \tag{S12}$$

so that

$$\begin{aligned} I_{in} &= -\frac{R_2}{R_1 R}(V_a - V_b), \\ I_{out} &= \frac{1}{R}(V_a - V_b). \end{aligned} \tag{S13}$$

When $R_1 = R_2$, we have

$$\begin{aligned} I_{in} &= -\frac{1}{R}(V_a - V_b) = -I_{out}, \\ R_{ab} &= -R = -R_{ba}. \end{aligned} \tag{S14}$$

From the equations, one can find that the effective resistance from A to B is $-R$, while from B to A is $+R$, thereby realizing the non-reciprocal resistance part. The resulting complex hopping provided by this functional unit is of the form $C_{ab} - \mathrm{i}/(\omega R_{ab})$, where the variable capacitor C_{ab} and the resistor R_{ab} provide independent control of the real and imaginary parts.

In practical realization(Fig. **S10**), the variable resistor R_{ab} can be implemented with an R-$2R$ ladder programmable resistor (e.g., Analog Devices, AD5270), offering 2^{10} digitized levels and a resistance upper limit up to 100 $k\Omega$, sufficient for most topological models. Non-reciprocal resistance is realized by a practical operational amplifier (Analog Devices, LT1363). For the circuit stability, two feedback resistors and capacitors $R_1, R_2 = 1k\Omega$ and $C_1, C_2 = 100\mathrm{pF}$ are utilized.

As demonstration, we realize a complex hopping term $e^{\mathrm{i}\pi/6}$ (i.e., $cos(\pi/6) + \mathrm{i}sin(\pi/6) = 0.866 + 0.5i$) between node A and node B. Extra grounded inductance, capacitance, and resistance at A and B are compensated so that the operating frequency is 200 kHz. The tunable components are set as $C_{ab} = 68.9\mathrm{pF}$ and $R_{ab} = 20\mathrm{k}\Omega$. We use LTspice to compute the voltages and currents, and compute the circuit Green's function as:

$$G = 1 \times 10^3 \times \begin{bmatrix} -0.0003 + 0.0028\,\mathrm{i} & -5.0147 + 8.6696\,\mathrm{i} \\ 4.9842 + 8.6563\,\mathrm{i} & -0.0239 + 0.3251\,\mathrm{i} \end{bmatrix} (\Omega).$$

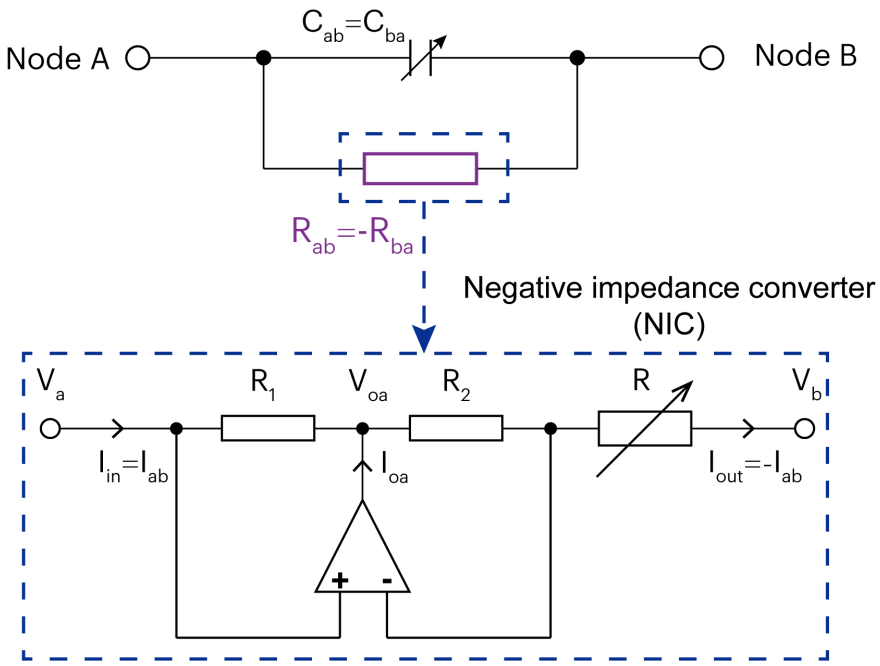

FIG. **S9**. **Hopping unit realizing complex value interaction.** Comprising the parallel of a tunable capacitor and a non-reciprocal resistor enabled by a negative impedance converter.

From which we obtain the Hamiltonian, for clarity, we omit the constant percentage coefficient that is identical in all nodes. Given H as:

$$H = \begin{bmatrix} -0.0325 - 0.0023\,\mathrm{i} & 0.8676 + 0.4996\,\mathrm{i} \\ 0.8643 - 0.4999\,\mathrm{i} & -0.0003 - 0.0000\,\mathrm{i} \end{bmatrix}.$$

The diagonal terms are onsite potentials, which are near zero at the central frequency. The hopping terms are non-reciprocal, and the complex values are quite precise to target $e^{\mathrm{i}\pi/6}$, the deviation is only 0.12%. Small discrepancies arise from the op-amp and parasitic effects.

Using the same procedure, we can vary the hopping term from $e^{i\pi/6}$ to $e^{i7\pi/6}$, the simulated results are shown in Fig. **S11**. Dots indicate the simulated hopping data result.

With this hopping unit introducing nonreciprocal complex couplings, topolectrical circuits can further emulate condensed-matter models that require breaking time-reversal symmetry, such as the Haldane model, thereby increasing the universality of our system. However, because the module uses active components, it introduces significant challenges for the system's complexity and stability. Here we model the feasibility of such a hopping unit, and we will continue to optimize the module and implement it for network architecture in future

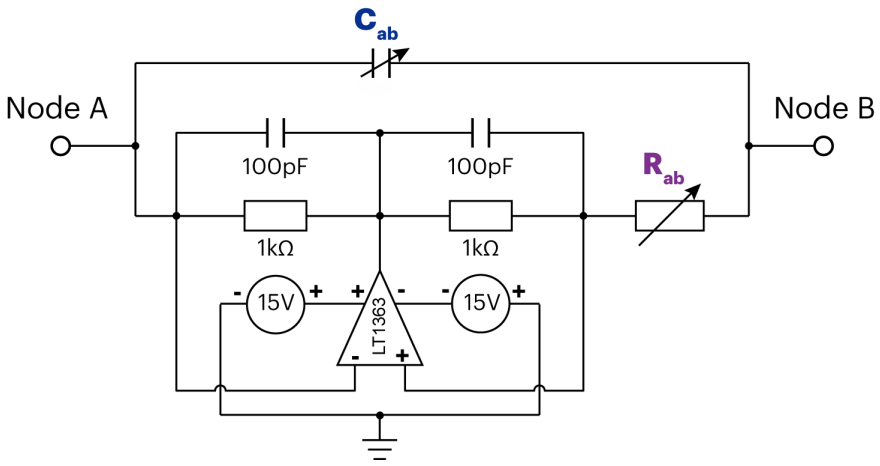

FIG. **S10**. **The schematic of practical hopping unit design.** C_{ab} and R_{ab} are the tunable capacitor and resistor that decide the real and imaginary part of the hopping term. Operation amplifier LT1363 is utilized to provide the active characteristic, and extra components serve as feedback to maintain the stability of the circuit.

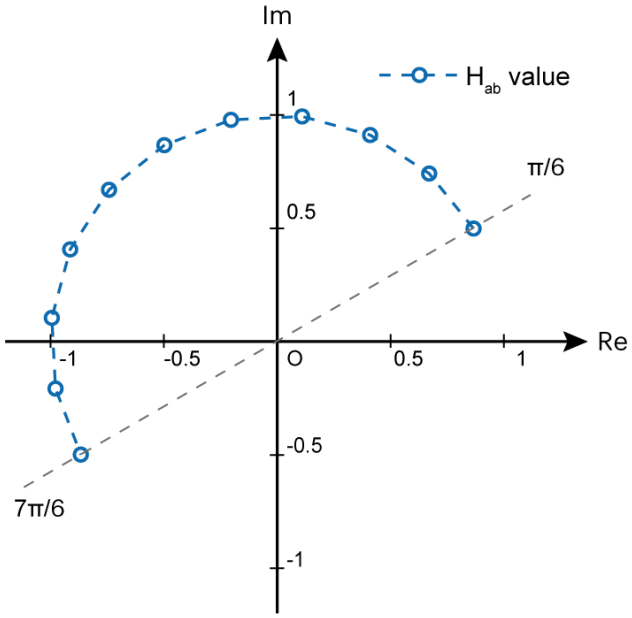

FIG. **S11**. **Hopping term phase in the range** $[\pi/6, 7\pi/6]$ **plotted in the complex plane.**

work.

It should be noted that these advanced hopping units are modular upgrades. They can be integrated by upgrading and replacing the existing programmable units, leaving the DLPTC architecture, AI control stack, and the scientific conclusions entirely unaffected.

S2. HIGHER-ORDER TOPOLOGICAL INSULATOR WITHOUT GLOBAL SYMMETRY ON PTC PLATFORM

A. The pictures of the ground states and analysis in two extreme couplings

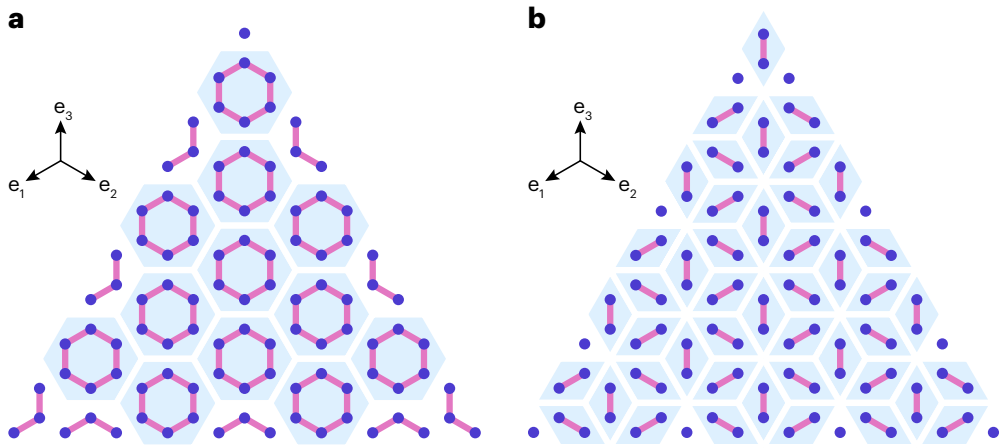

FIG. **S12**. **Ground states in HOTI: (a)** $t_1 = 0, t_2 = 1$**. (b)** $t_2 = 0, t_1 = 1$**.**

To analyze the topological properties, we tune the weak coupling coefficient to zero. While this process is adiabatic (i.e., the band gap remains open and the topological invariants remain the same), it provides a way to a direct understanding of the topological states. Figure **S12**a shows the case $t_1 = 0, t_2 = 1$, and Fig. **S12**b shows the case $t_2 = 0, t_1 = 1$. In case (a), monomers, trimers, and hexamers exist [2], and the system has seven energy levels $E \in \{0, \pm 1, \pm 1.5, \pm 2\}$. Monomers at the corner and trimers at the edge position contribute to the $E = 0$ level, constituting the bound corner in the edge continuum (BEIEC) state. It means that the corner states are observed to reside within the edge band. Hexamers, contributing the energy levels $E = \pm 1, \pm 2$, constitute the bulk state. The energy at the $E = \pm 1.5$ level, the modes are exclusively located in trimers at the edge boundary, constituting the edge modes. In case (b), monomers and dimers exist, and the system has only three energy levels $E \in \{0, \pm 1\}$. Monomers, contributing zero-level energy, all situate at the edge boundary and signify the presence of edge modes. Dimers, contributing to the $E = \pm 1$ levels, encompass two corner BIC modes alongside most bulk modes. Figs. **S13**, **S14**, **S15**, and **S16** show the degenerated modes which energy $E \in \{0, \pm 1, \pm 1.5, \pm 2\}$ of case(a). Figs. **S17**, and **S18** show the degenerated modes which energy $E \in \{0, \pm 1\}$ of case(b).

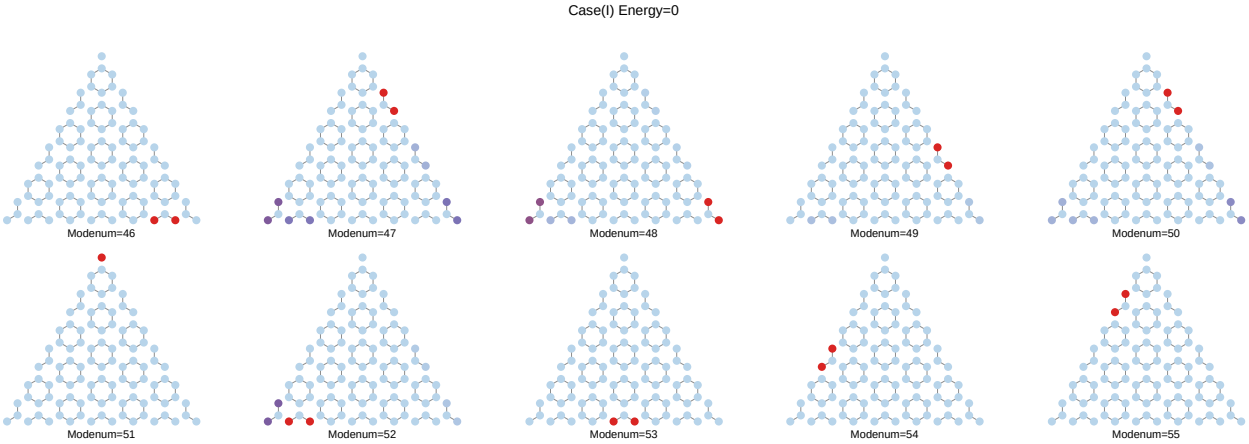

FIG. **S13**. **Modes in case(a) with Eigenenergy=0.**

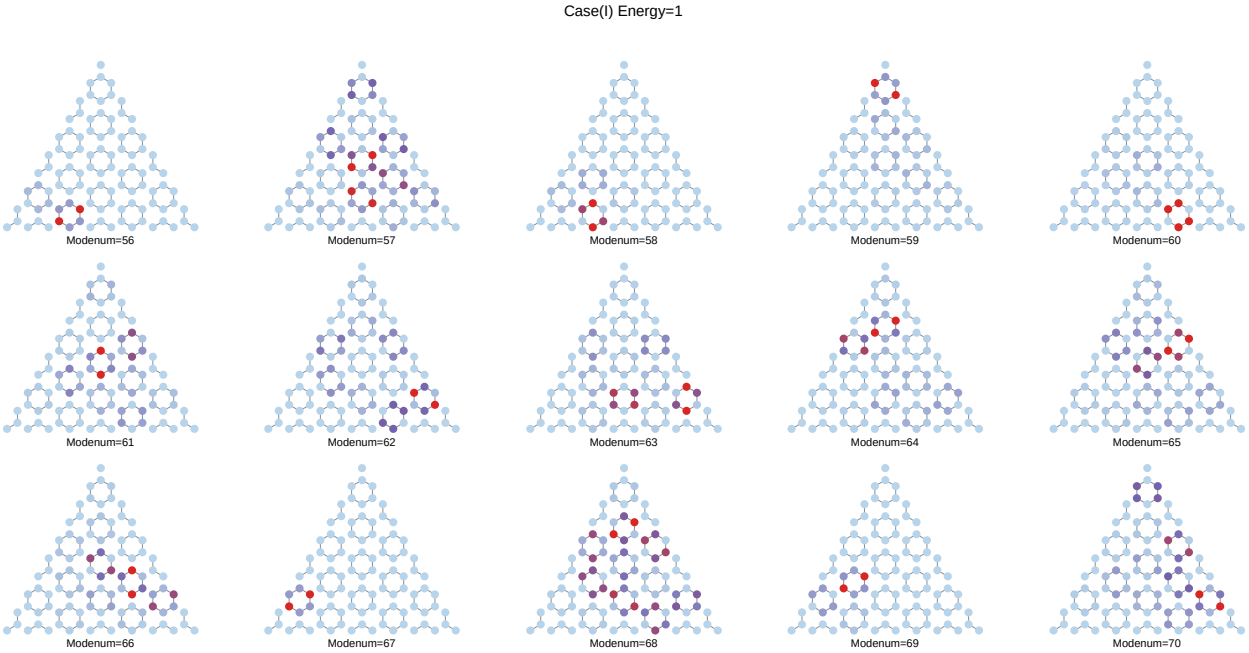

FIG. **S14**. **Modes in case(a) with Eigenenergy=±1.**

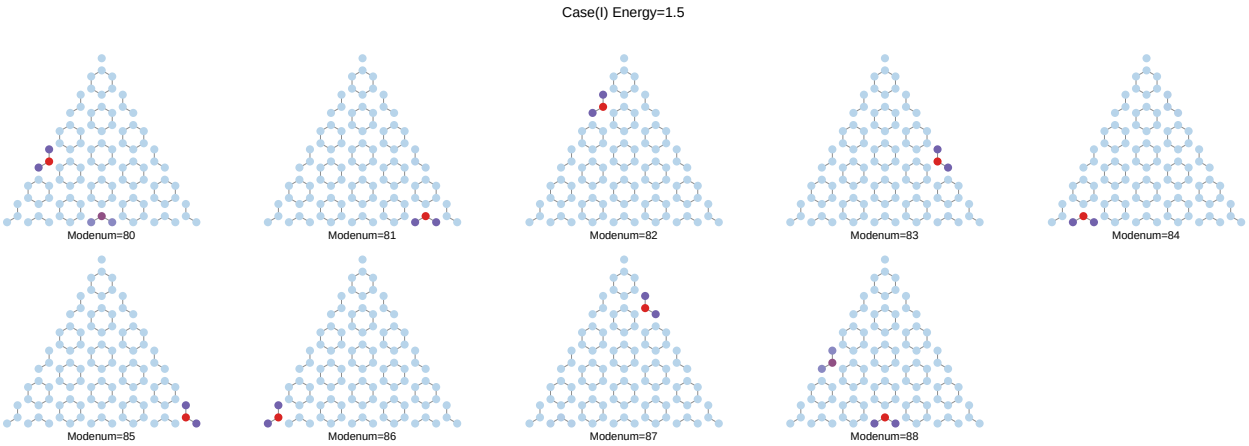

FIG. **S15**. **Modes in case(a) with Eigenenergy=±1.5.**

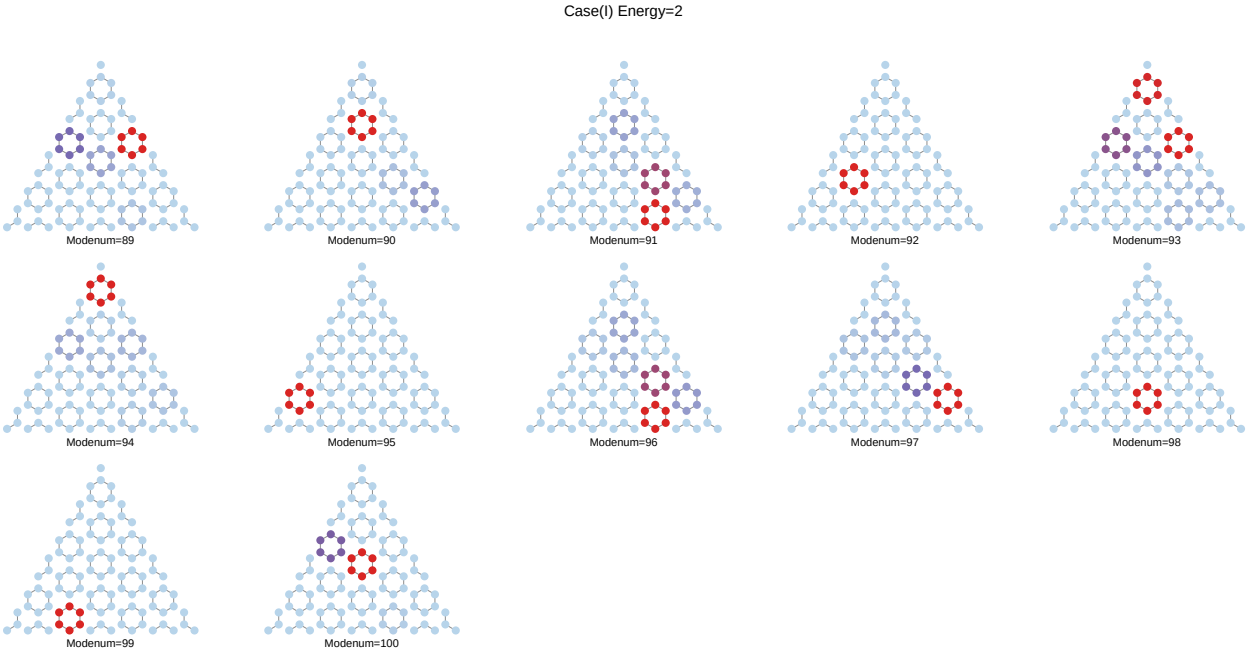

FIG. **S16**. **Modes in case(a) with Eigenenergy=±2.**

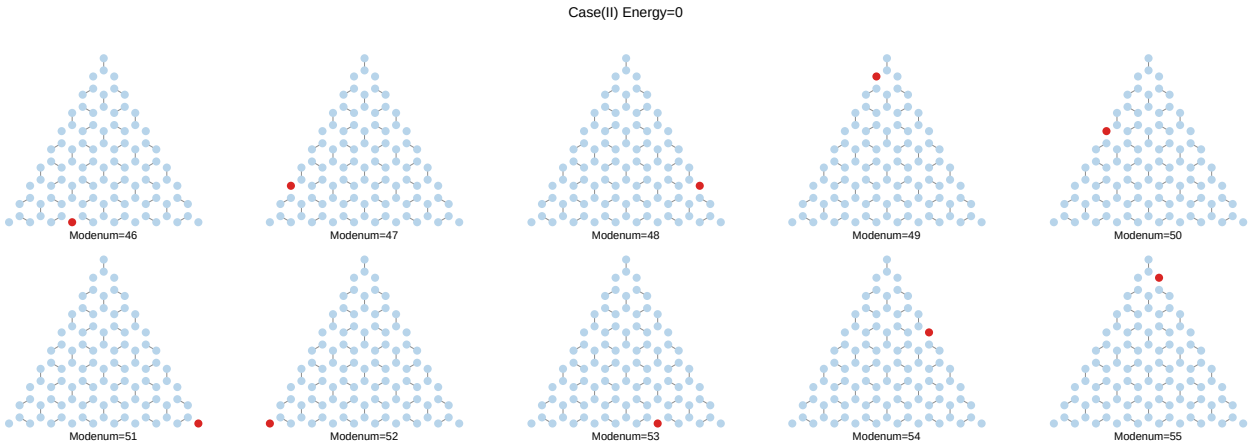

FIG. **S17**. **Modes in case(b) with Eigenenergy=0.**

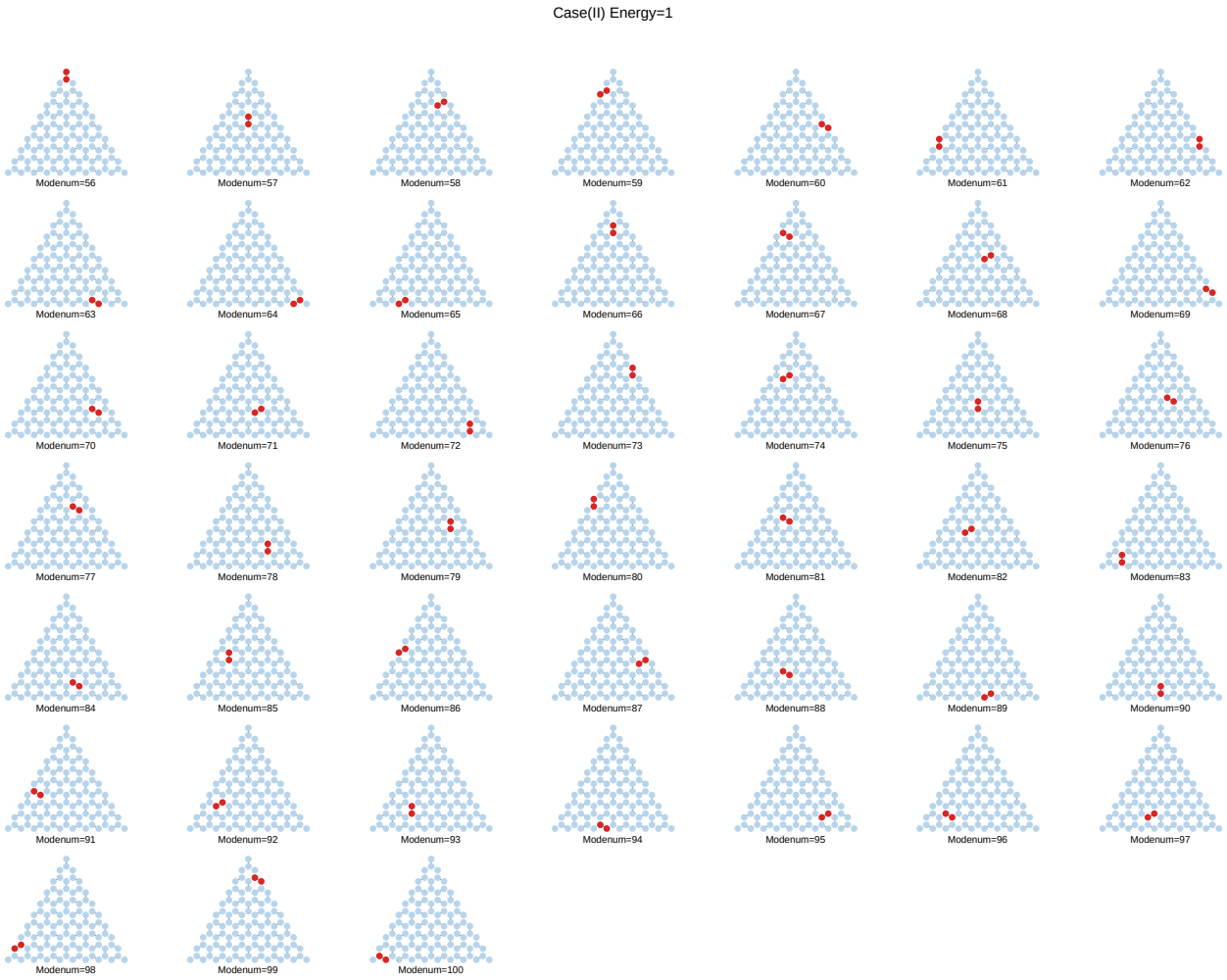

FIG. **S18**. **Modes in case(b) with Eigenenergy=±1.**

B. Theory of HOTIs without global symmetry

To illustrate the novel topological phase transition, the theory in [3–5] analyzes the distribution of the Wannier centers and explains the topological origin of HOTIs via the fractional charges ("filling anomaly" in the terminology of solid-state physics). Usually, topologically phases of HOTIs are associated with the unit cells of a periodic lattice with global symmetry, while the truncation of the lattice keeps the number of unit cells as an integer. The occurrence of fractional charges can be visualized by integrating the local density of states in the occupied band per unit cell, as confirmed by the calculation of the density of states.

In the Figs. **S19**a1-3, the Wannier centers are located in topologically trivial positions, the associated mode density is restricted to the unit cell, and no fractional charge can arise when the lattice is truncated. In the Figs. **S19**b1-3, the edge Wannier centers are located in the topologically nontrivial positions, the associated mode density is distributed differently in the boundary unit cells, and the fractional charge occurs when the lattice is truncated. Thus, in HOTIs with global symmetry containing an integer number of unit cells, the topological phase transition can be controlled by the dimerization of the intracell (t_1) and intercell (t_2) coupling strengths, resulting in a topological phase with corner modes at $t_2 > t_1$ and a trivial phase corresponding to $t_2 < t_1$.

In contrast, we present an HOTI without global symmetry, which contains a structure with *complete* and *incomplete* unit cells, and show that the higher-order topologically protected mechanism exists in it in both $t_2 > t_1$ and $t_2 < t_1$ regimes. Unlike HOTIs with global symmetry, where the topological nontrivial phase only occurs at $t_2 > t_1$, in HOTIs without global symmetry, boundary states are found both at $t_2 < t_1$ and $t_2 > t_1$. There is still a transition point $t_2 = t_1$, which distinguishes between case I and case II phases, which are shown in Figure 2 of the main manuscript.

Figures **S19**c1 and d1 show the positions of the Wannier centers. In case I, the Wannier centers are located at the topologically trivial positions (centers of the unit cells), while truncation leads to the appearance of *fractional Wannier centers* in the corners and at the edges of the lattice. In case II, the Wannier centers are located at the topologically nontrivial positions (boundaries of the unit cells), while truncation leads to the appearance of *fractional Wannier centers* only at the edges of the lattice. This implies that even if the truncation occurs through the interior of several unit cells, there is still a topologically

protected mechanism supporting the existence of localized modes.

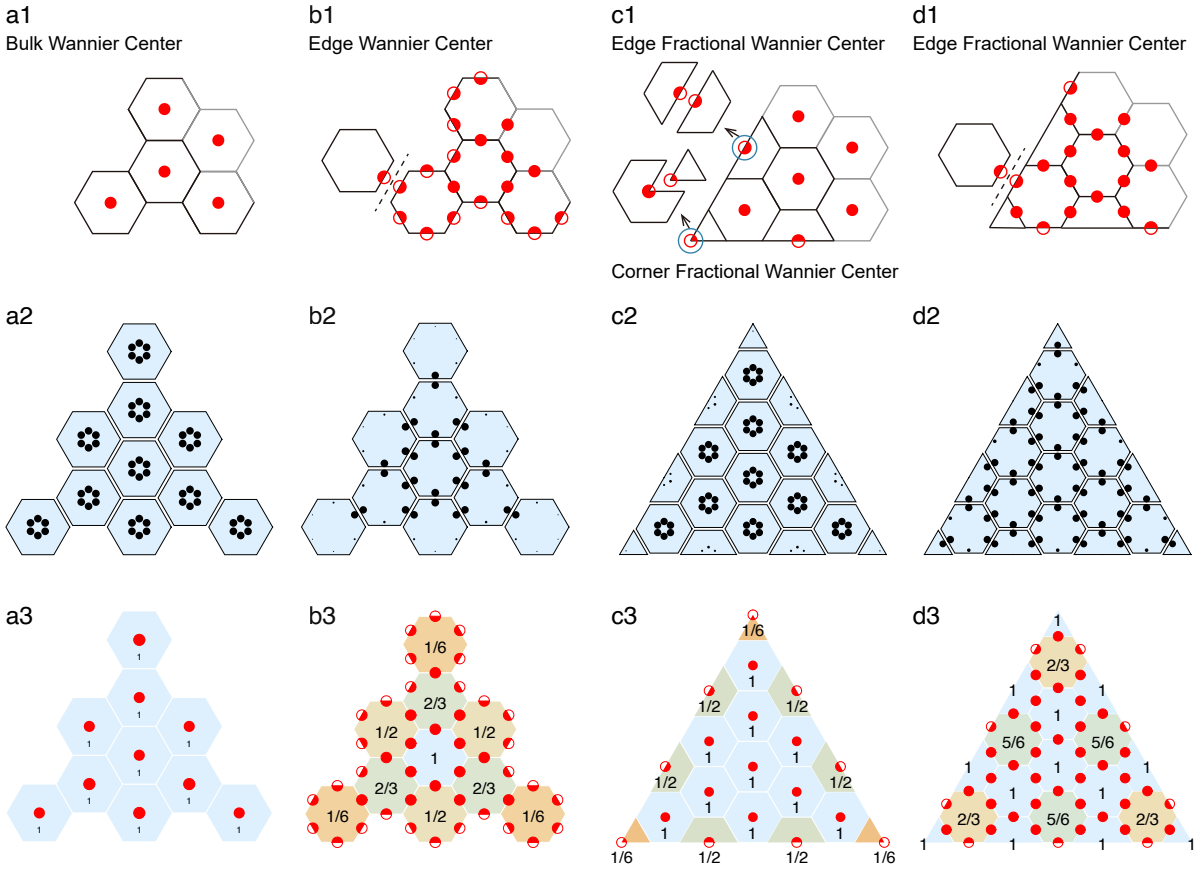

FIG. **S19**. **Schematic representation of the concept of fractional Wannier center, distributions of mode density, and corresponding fractional charges.** (a,b) The concept of Wannier centers, distributions of mode density, and corresponding fractional charges in topological trivial phase ($t_2 < t_1$) and topological nontrivial phase ($t_2 > t_1$) in HOTIs with global symmetry. (c,d) The concept of Wannier centers, distributions of mode density, and corresponding fractional charges in case I ($t_2 < t_1$) and case II ($t_2 > t_1$) in HOTIs without global symmetry.

To generalize the concept of fractional charge, we consider the incomplete unit cells as if they were complete, with completely filled additional imaginary vacant areas. In Fig. **S19**c3, the fractional charges for corner zones and edge zones are given by 1/6 and 1/2, respectively. In Fig. **S19**d3, the fractional charges for edge zones are equal to 2/3 and 5/6.

C. Phase transition diagram via $M = t_1/(t_1 + t_2)$

In case I of the last section, the topological system is reduced to a set of isolated hexagons in the bulk region and a set of monomers, trimers at the boundary, while it is reduced to a set of isolated dimers in the bulk region and a set of monomers at the boundary for case (b). These ground states contribute different system states in topological systems, corresponding to $M = 0$ and $M = 1$. When $M \in (0, 1)$, modes are varied and phase transitions happen. The system Hamiltonian of each condition M has 100 eigenmodes with lots of degenerated eigenenergies, labeled from mode 1 to mode 100 by sorting the eigenenergies from low to high. When M is increased from 0 to 1, case(a) is gradually transformed to case(b). The edge states observed in case(a) have transformed into bulk states in case(b). Notably, two specific edge states, designated as mode 17 and mode 84, have transitioned into corner states, which subsequently become corner-bound states in the continuum (Corner BIC), as labeled on the lines with a gradient ramp. Furthermore, the BCIEC states that with zero energy are transitioned into edge states. It can be observed that, divided by the $t_1/(t_1+t_2) = 0.5$, there are phase transitions that happen. Figures **S20**, **S21**, and **S22** illustrate the HOTI phase transition of the edge states to bulk states. Note that among them, mode 17 in Fig. **S21**, is the transition of edge state to corner BIC state. Figure **S23** shows the phase transition from edge states to bulk states, and Fig. **S24** shows the phase transition from BCIEC states to edge states.

D. Circuit implementation of HOTIs without global symmetry

Considering the Hamiltonian and circuit parameter, the capacitance for weaker coupling is set to 30 pF and the strong coupling is set to 90 pF so that the hopping parameters can be transitioned from $t_1 : t_2 = 1 : 3$ to $t_1 : t_2 = 3 : 1$ (M from 0.25 to 0.75). The on-site capacitance is also precisely configured to make all on-site potentials identical. The central frequency is tuned to 265.26 kHz.

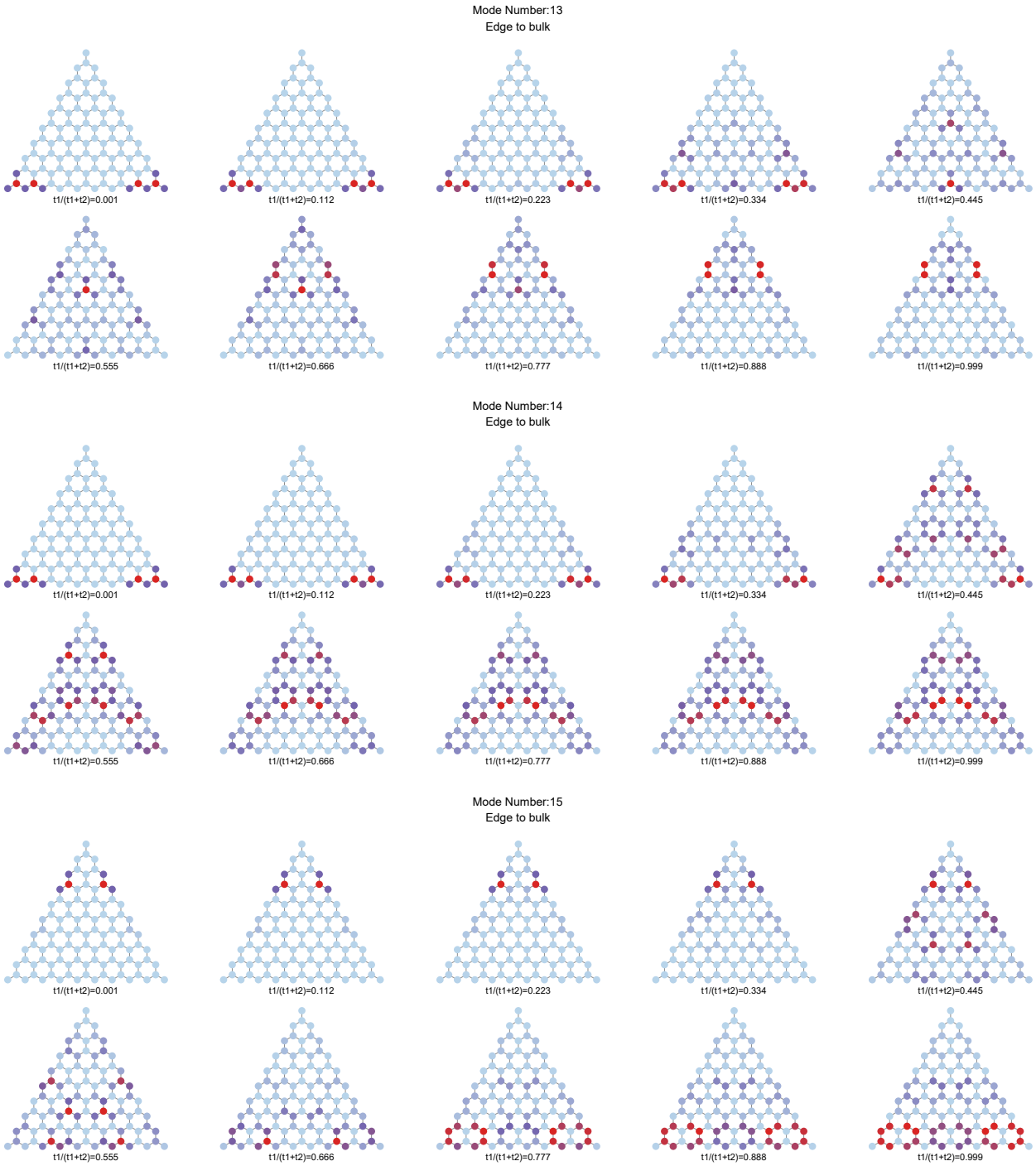

FIG. **S20. Phase transition from edge state to bulk state: mode 13-15.**

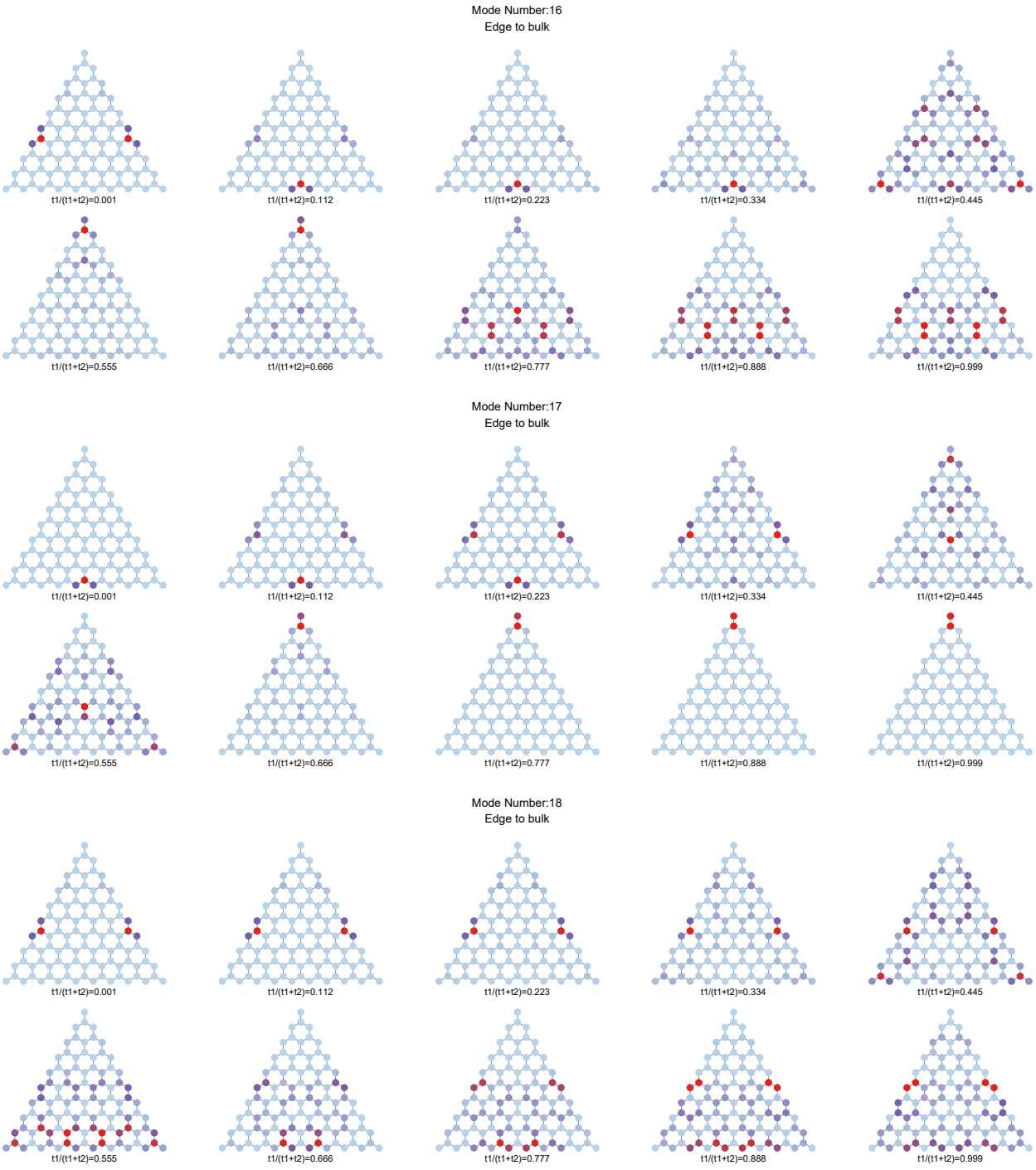

FIG. **S21. Phase transition from edge state to bulk state: mode 16-18.**

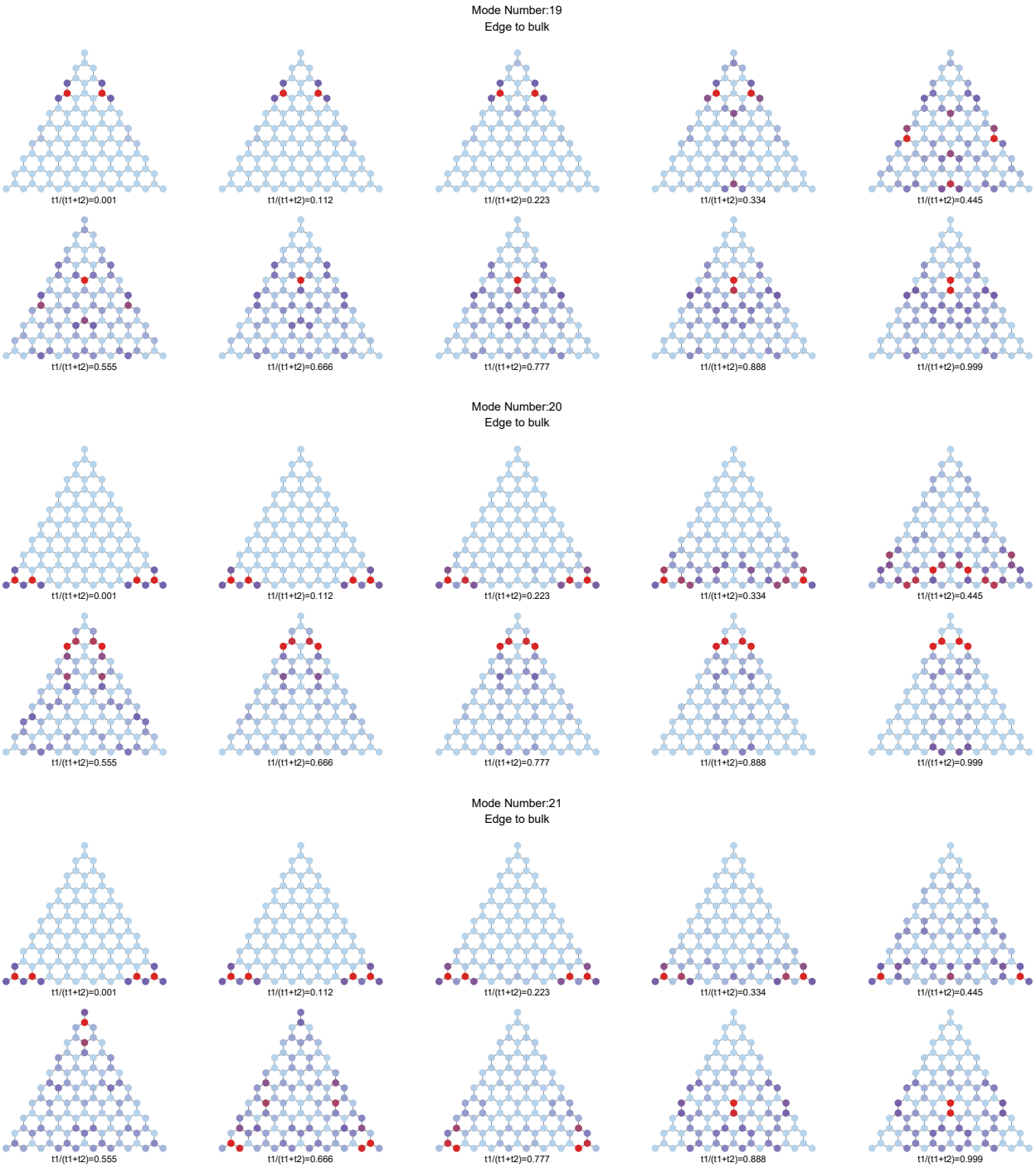

FIG. **S22**. **Phase transition from edge state to bulk state: mode 19-21.**

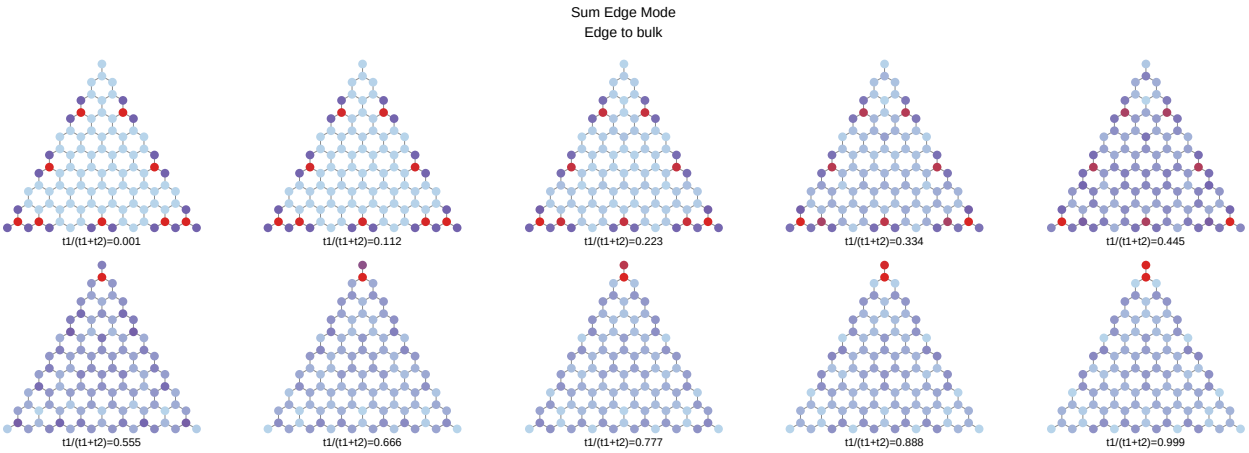

FIG. **S23**. **Phase transition from edge states to bulk states: mode summation of degenerate mode 13-21.**

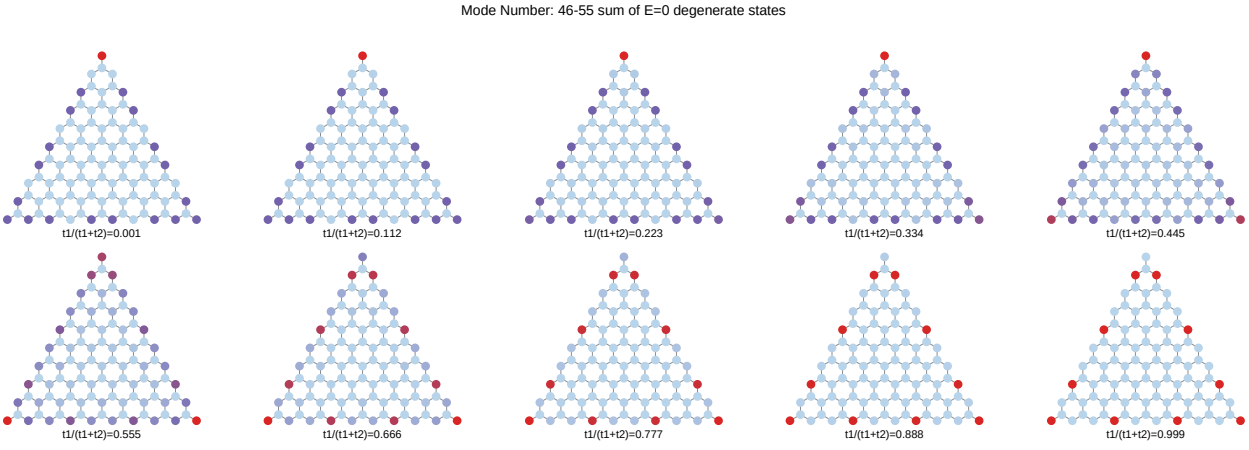

FIG. **S24**. **Phase transition of BCIEC states to edge states: mode summation of 10 degenerate modes 46-55.**

E. Calculation of the filling anomaly using the density of states

To calculate the filling anomaly, we use the retarded Green function $G_R(E) = \lim_{\varepsilon\to 0^+}(E + \mathrm{i}\varepsilon - H)^{-1}$, where H is the Hamiltonian. $G_R(E)$ defines as a meromorphic function of the parameter E . We define $G_R(r, r'; E) = \langle r\,|G_R(E)|\,r'\rangle$ with r denoting the position. The local density of states is given by $\rho_r(E) = -\frac{1}{\alpha}\mathrm{ImG_R(r, r; E)}$ with the normalization factor $\alpha = -\mathrm{Im}\int \mathrm{G_R(r, r; E)dE}$. The local density of states corresponding to the filling factors E_a and E_b is:

$$\rho_r(E_a, E_b) = -\frac{1}{\alpha}\mathrm{Im}\int_{\mathrm{E_a}}^{\mathrm{E_b}} \mathrm{G_R(r, r; E)dE}. \tag{S15}$$

In a circuit, the current I_r flowing into site r at frequency f follows Kirchhoff's law:

$$I_r = \sum_{r'} J_{r,r'}(2\pi f)V_{r'}, \tag{S16}$$

where $V_{r'}$ is the voltage at site r'. Considering that the ratio of the coupling capacitors equals the ratio of the coupling constants in the tight-binding model, we have:

$$J(2\pi f_0) = \mathrm{i}2\pi f_0 C_1 H. \tag{S17}$$

We calculate the filling anomaly by plugging $H = \mathrm{Im}J(2\pi f_0)$ into Equation S15. In the circuit parameters, the impedance is dominated by the smallest eigenvalue $j_n(\lambda, \omega)$ of $\mathbf{J}(\lambda, \omega)$ at a given frequency and maps to the local density of states (LDOS) in the tight-binding model. In this way, by measuring the impedance curve in the frequency range, we can derive the LDOS of the system.

S3. FLAT-BANDS AND LANDAU LEVELS ON PTC PLATFORM

A. Circuit implementation of all-band-flat model

To realize all-band-flat (ABF) on our PTC platform, we precisely control the coupling strengths. For the model scale $N = 9$, the gradient proportion of hopping terms is varied from $\sqrt{1}$ to $\sqrt{9}$. Considering the circuit characteristics, the minimum intensity unit of hopping terms is selected to 30 pF, and all hopping terms are programmed to precisely control the coupling strengths to realize all-band-flat (ABF) on our PTC platform for $30\times(\sqrt{n} = \sqrt{1}, \ldots, \sqrt{9})$ pF. The grounded inductances in parallel with all sites are identical to 1.5 mH. The capacitance of on-site potential terms is also finely controlled so that the total capacitance on each site remains identical to 196.96 pF.

B. Simulated and measured DOS spectra with parasite parameter

Simulation can perform circuit components with no parasitic parameters. An ideal LC resonator constituted by capacitors and inductors without any parasite parameters and no resistance has an almost infinite quality factor, which induces a δ-function shape-like resonance peak. With a very narrow bandwidth and almost infinite impedance, it needs a very high simulation resolution to capture the resonance peak. As a result, we demonstrate two cases in simulation: I) a model with small enough parasite parameters, to keep the finite quality factor, and illustrate all phenomena that the theoretical model predicted; II) a model with actual parasite parameters in device components, to compare with the experimental result.

The impedance curves at each node are calculated and summarized to derive the overall impedance spectrum, indicating the DOS spectrum, as shown in Fig. **S25**. From the impedance spectrum in case I, all 19 bands can be observed. However, due to the spectrum broadening caused by inevitable parasite parameters (Equivalent Series Resistance, Equivalent Series Inductance) of capacitors and inductors, partial peaks at the lower frequency side overlap.

The impedance curves at each node of simulation cases I, II, and experimental measurement are shown below. We show only 22 characteristic nodes, and the remaining nodes have a rotation symmetry with respect to the geometry, so their value is identical to one of the 22 nodes. The selected region and its site labels are shown in Fig. **S26**.

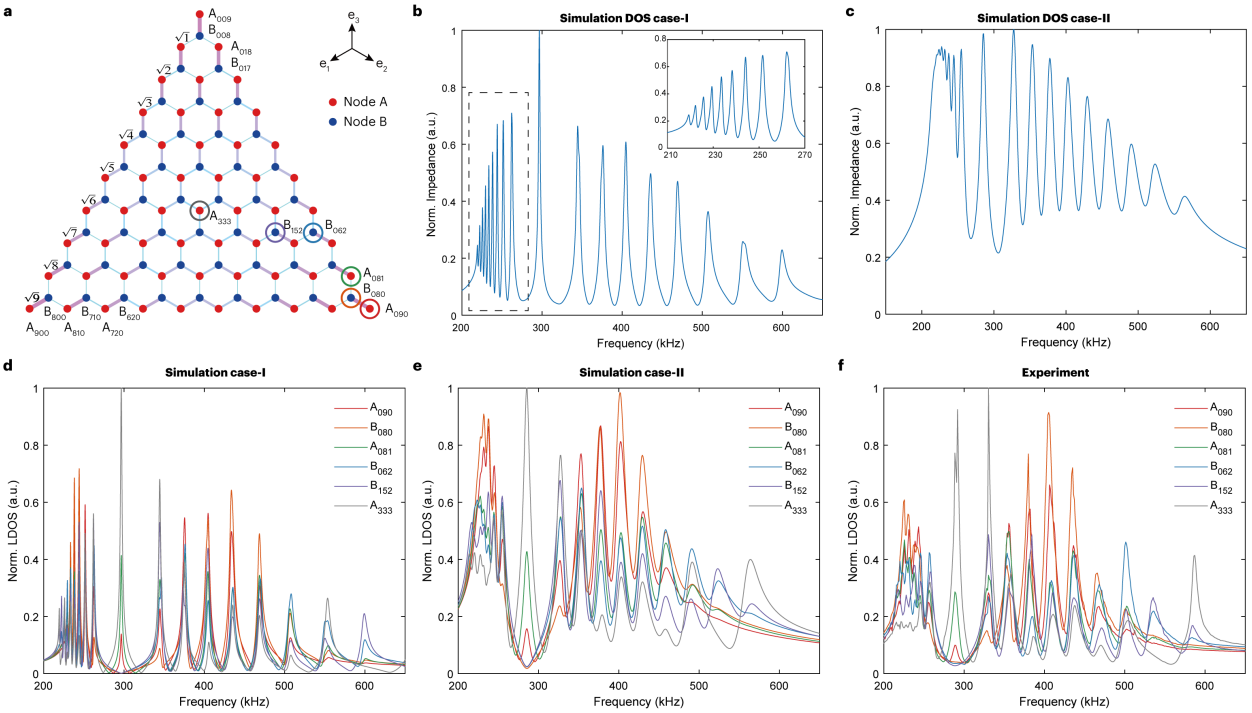

FIG. **S25**. **Circuit system and overall impedance spectrum.** (a) Schematic of the all-flat-band lattice. (b) Impedance spectrum in case I. (c) Impedance spectrum in case II. (d)-(f) Impedance spectra of each point shown in **(a)** for case I, case II and experiment.

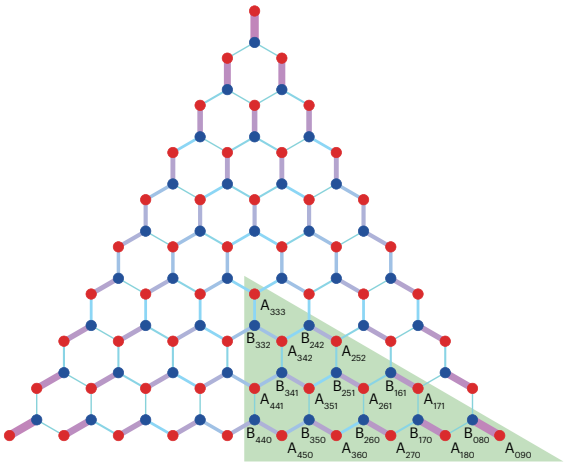

FIG. **S26**. **The selected region and its site labels in all 100 sites.**

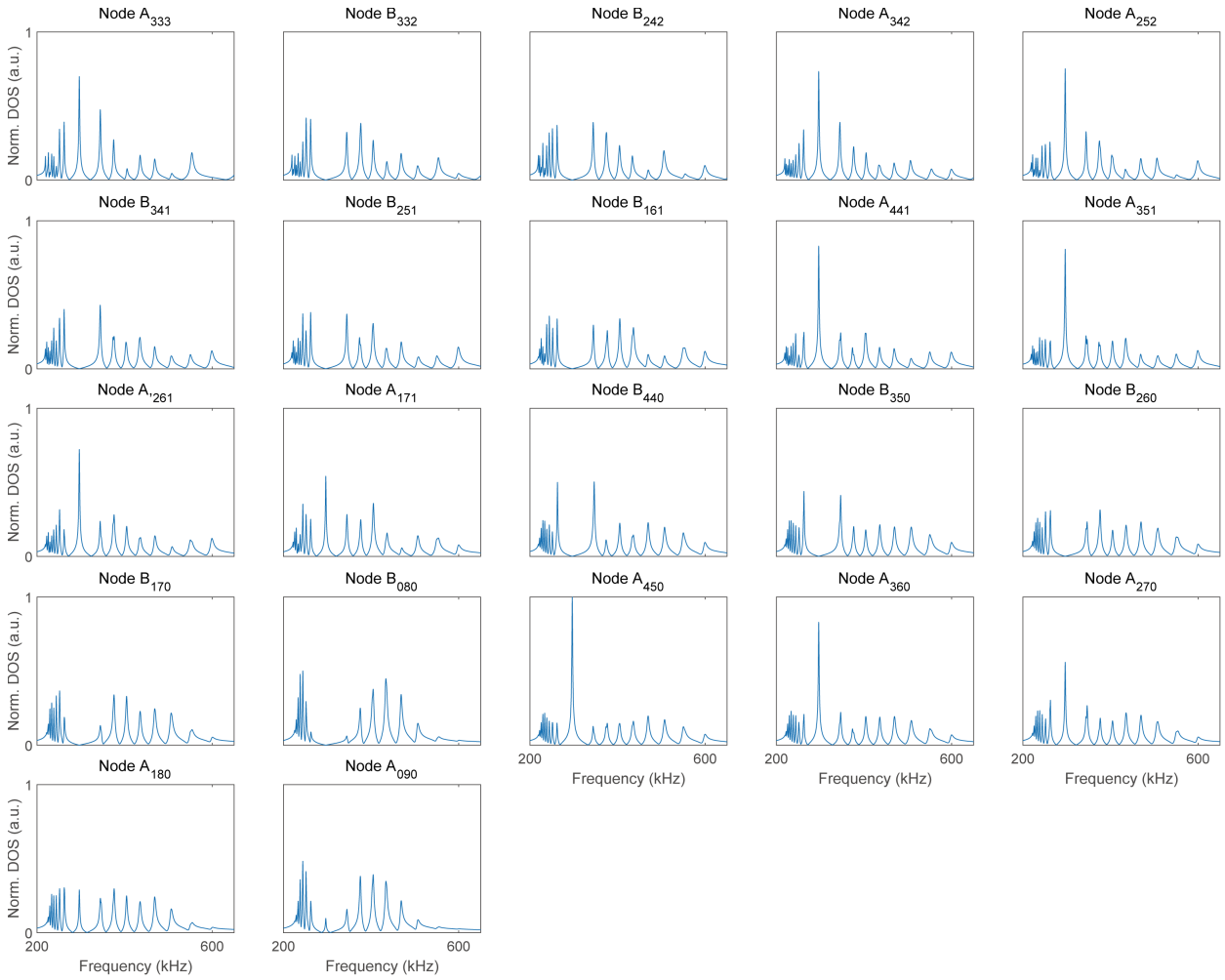

FIG. **S27**. **The simulated impedance curves in case I.**

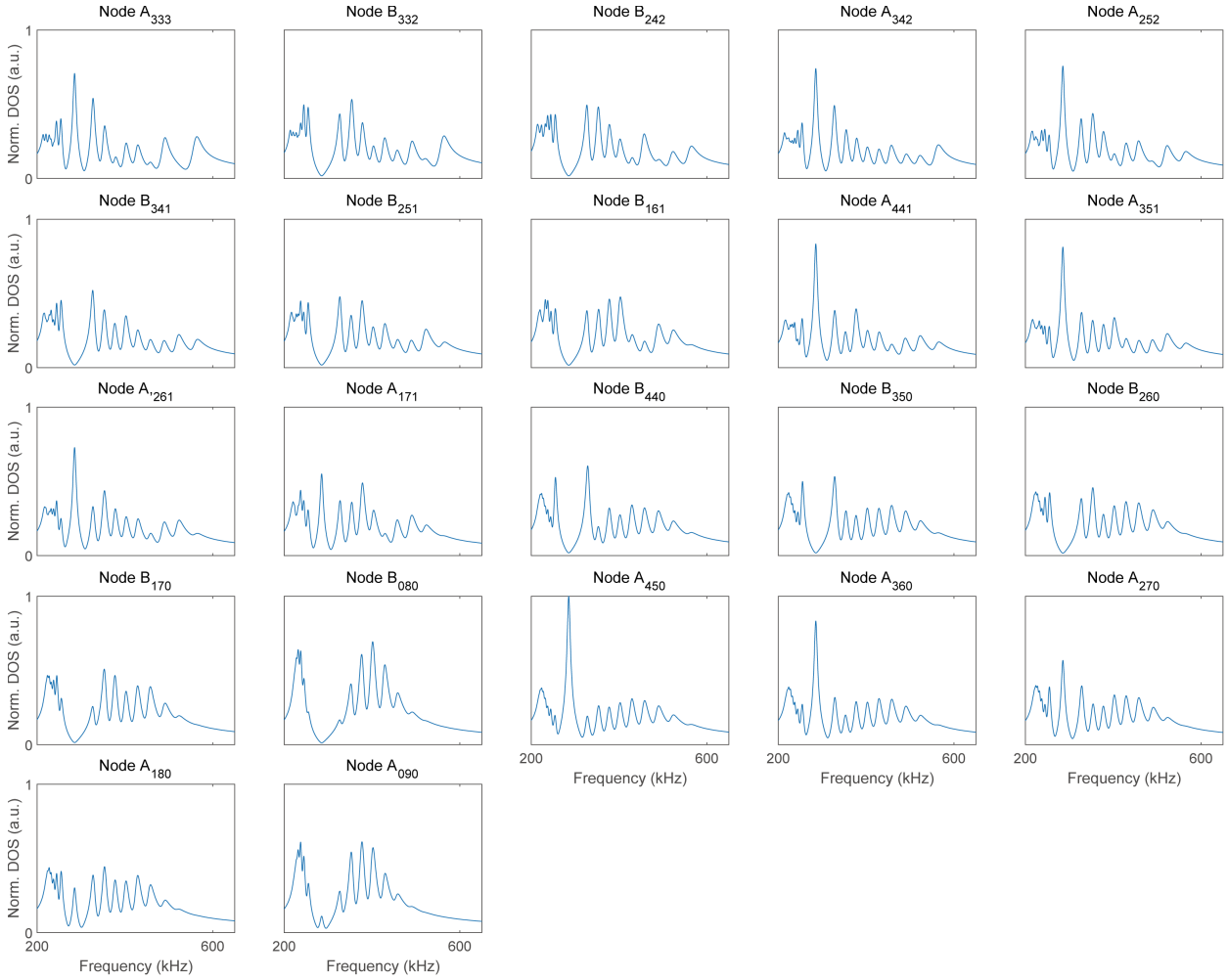

FIG. **S28**. **The simulated impedance curves in case II.**

FIG. **S29**. **The experimental impedance curves.**

C. Mode distribution for different Landau levels

As can be observed in the mode distribution (Fig. **S30**), the pattern of the zeroth Landau level M_0 shows anomalous parity, manifesting as nonzero LDOS only in the sub-lattice at Node A. For the higher-order Landau level from M_1 to M_4, we observe that the mode functions spread from the center to the edges when mode order increases, and then shrink to the center when continuously increasing from M_5 to M_9. During this "breathing" process, their spatial distributions remain C_3 symmetry with respect to the center of the lattice [6].

Especially, there are many distinctive characteristics in these modes. For example, on M_1, the eigenmodes on the A sites have high weights in the three corners of the lattice, while for M_6, the three edges are preferred. The 8th Landau level M_8 has an annulus distribution with zero intensity in the center, and the M_9 Landau level, which contains only one mode, is localized near the center. The distinctive distributions of the Landau levels give us an additional controlling knob to selectively excite a Landau level at specific positions of the lattice.

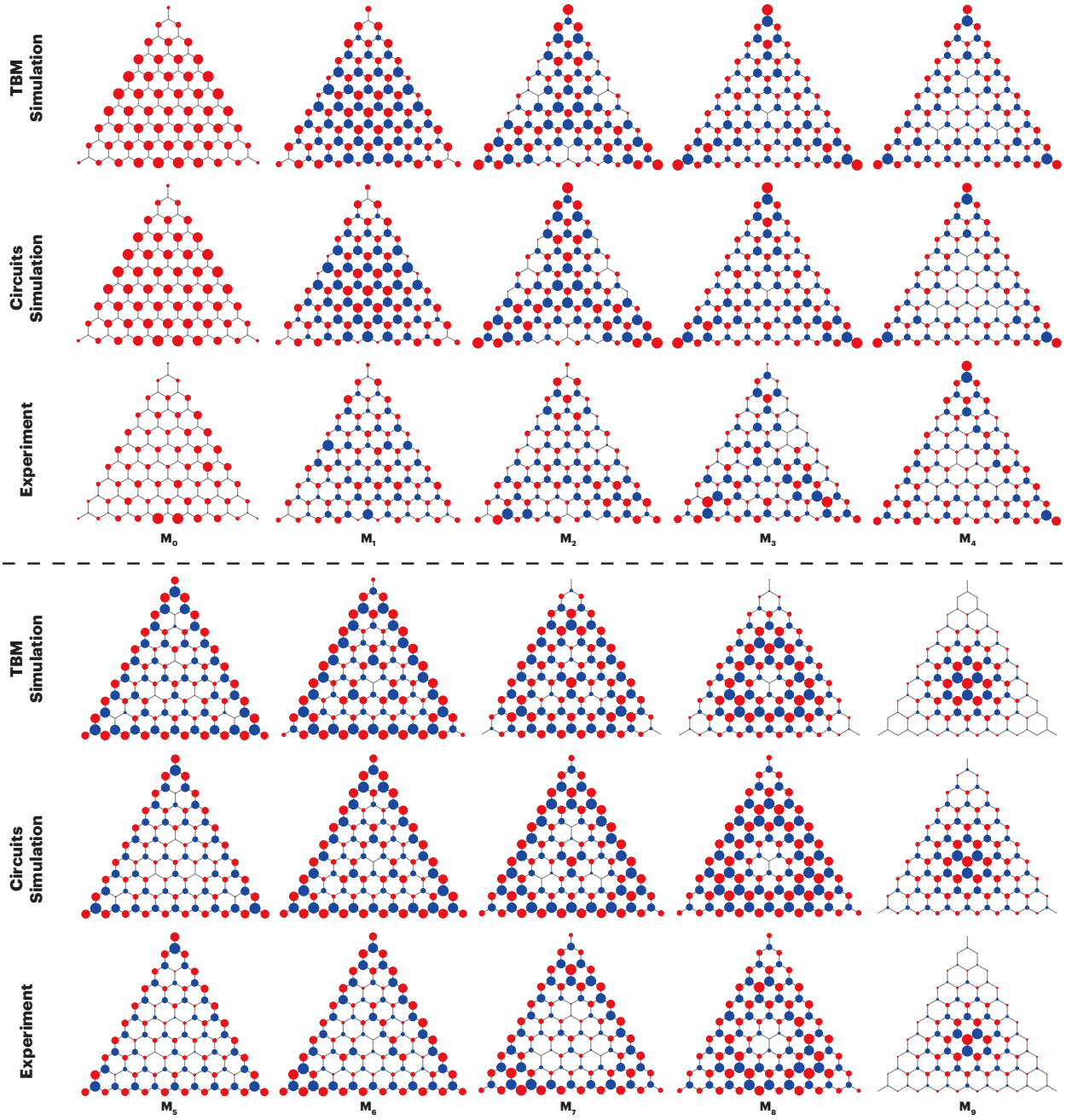

FIG. **S30**. **The mode distribution of modes M_0 to M_9.**

S4. DEEP-LEARNING EMPOWERED PROGRAMMABLE ANDERSON LOCALIZATION

A. Physics-informed data screening method for high-performance localization

The tight-binding model, which can fully describe the topological properties and determine wave functions, serves as the physical engine to calculate the physical characteristics (localization performance, etc.) of a Hamiltonian.

As we discussed in the article, the system is described by $G = [\mathrm{rand(V)}, \mathrm{E}]$, where rand(V) represents the on-site random disorder terms. In the circuit platform, the wave function distribution can be detected using the grounded impedance distribution at a certain excitation frequency. The impedance on each point k can be written as:

$$Z_k = \frac{\sum_{m=1}^{100} |\psi_m(k)|^2}{j_m} \tag{S18}$$

meaning that it is decided by all wave-functions $\psi_m(k)$ at point k, and weighted by the inversion of eigenvalue j_m. It is dominated by the wave function with the smallest eigenvalue ($j_m \approx 0$) [7]. The phenomenon of Anderson localization indicates that energy is primarily concentrated in small regions of a physical system. This is evidenced by the impedance peak observed in the corresponding small region.

To provide a quantitative indicator of localization degree for physics screening, the inverse participation ratio (IPR) parameter is introduced [8]:

$$\mathrm{IPR} = \frac{\sum_{i=1}^{N} |\psi(r_i)|^4}{\left(\sum_{i=1}^{N} |\psi(r_i)|^2\right)^2}. \tag{S19}$$

where $\psi(r_i)$ is the wave function of certain modes under randomly disordered onsite terms. Here it can be derived by the weighted all eigenmodes as Eq. (S18), or approximated by using the eigenmodes with the smallest eigenvalue instead.

The physics-informed data screening method utilizes the provided physics-graph information to filter vertex value vectors that satisfy the criteria for localization mode alignment with the IPR constraints, localization position, and intensity specifications. The implementation of the physics-informed data screening process is delineated below, as Eq. (S20). For a set of randomly generated disorder vectors denoted as rand(V), the method first solves

the Hamiltonian using the TBM method to obtain the wave function $\psi(r)$. Subsequently, it identifies the position of the maximum of $\psi(r)$ and calculates the IPR value based on Eq. S19. Then, by filtering the vectors, it retains only those with IPR > threshold that match the target localization position, as depicted in Fig. **S31**a.

$$\begin{aligned} \text{Randomly sample } \mathbf{V} &= (V_1, V_2, \ldots, V_{100}) \sim U(-\frac{S}{2}, \frac{S}{2}), \\ \text{Subject to } [\psi(r), E] &= \text{eig}[H + \text{diag}(\text{Vector})], \\ \text{IPR} &> \text{threshold}, \\ \arg\max\left(|\psi(r)|^2\right) &= \text{Target Position}. \end{aligned} \tag{S20}$$

By searching the solution space with a large amount of rand(V), the derived $V_{selection}$ is used as training data and fed into the neural network. We filter the results with IPR > 0.3, therefore, the intensity of the center point of the localized region I_{center} is also large enough for information encoding. Under such localization strength, the process from a random vector to a certain localized position becomes a multiple-to-one mapping. Due to the selection of random disorder value being infinite, there will be an infinite number of solutions for each localized position (Fig. **S31**b).

For example, a collection of these random disorder vectors converges on the same localized site position '30'. By circuit simulation and implementing the parameters to the PTC platform, the results can be observed that the energy is precisely localized to the designed position, shown in Fig. **S31**c, which also matches well with the simulation result. The Anderson localization position is used as an information label for the dataset.

B. Symmetry-enhancement for training dataset preparation

Data augmentation is an effective methodology employed in the preparation of training datasets [9], aimed at artificially increasing both the size and diversity of the dataset by creating modified versions of existing data. This approach enhances the performance and generalization capabilities of the deep learning models. It holds particular significance for topological structures whose topological features are well known to be constrained (and in certain instances determined) by the inherent symmetry properties. Based on the physics symmetry of our system, the minimum area without repetition is shown in the origin of Fig. **S32**. We have the following data transforms:

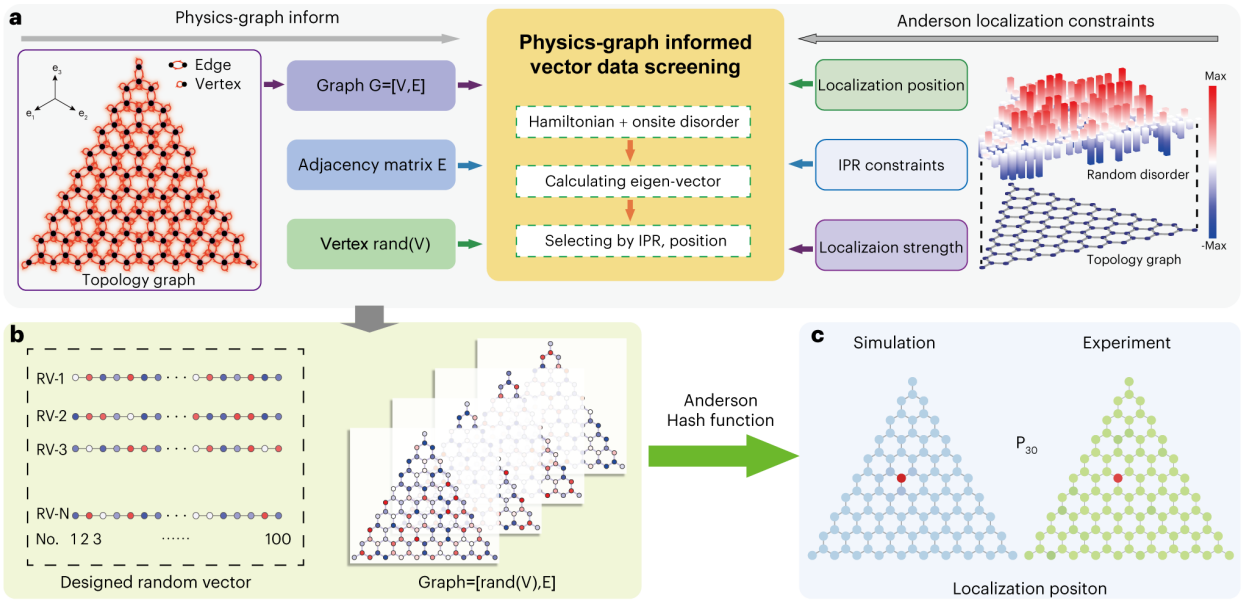

FIG. **S31**. **Data screening method for dataset preparation and validation of localization phenomenon.** (a) Data screening pipeline. Using the physics-graph information and constraint conditions, data that can induce suitable physics characteristics can be screened. (b) The phenomenon that infinite solutions can be made for each localized position label, so that Anderson Hash function can be established. (c) Simulation and experiment results for localization on label '30'.

1. Axis symmetry: There are three kinds of axis symmetry that are rotated by 120°, as the Axis symmetries 1, 2, and 3 in Fig. **S32**. When disorder data are exchanged along such an axis symmetry, the localized mode and the site label are also exchanged through axis symmetry.

2. Rotation symmetry: There are two kinds of rotation when the disorder data is rotated by 120°, the localized mode, and the site label are also exchanged through rotation symmetry.

By combining the two kinds of data transformation, the original dataset is enhanced to six times of original number, from 120000 to 720000. Such data argumentation can enhance the learning capability of neural networks by introducing the intrinsic symmetry of topological graphs.

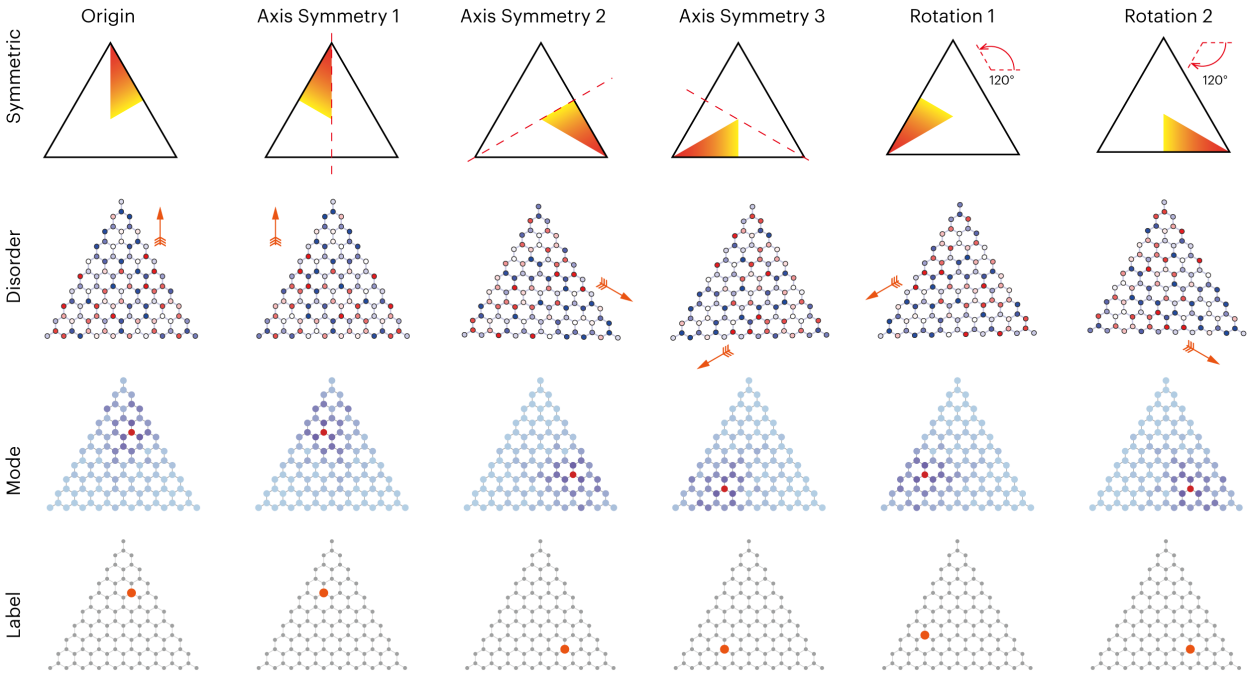

FIG. **S32**. **Data argumentation process.** Three-axis symmetries and two rotation symmetries are included.

C. The physics-graph-informed convolution neural network classifier design and optimization

The most commonly used neural network is the fully connected neural network (FCNN) [10], also known as the Multi-Layer Perception (MLP). Although it can theoretically approximate arbitrary functions given a sufficiently large network, its performance in our physics system is inadequate. Specifically, the disorder on-site vector rand(V) needs to be flattened before being transmitted into the FCNN. Such input data does not incorporate any graph-related or connectivity information, leading to a loss of essential physical information during the data input phase.

To address such drawbacks, we develop the physics-graph-informed convolutional neural network (PGI-CNN) in this article. The disorder vector rand(V) is combined with physics-graph information, meaning that the connection information is taken into consideration. The rand(V) as a $1\times$ 100 vector transformed into a 19 $\times$ 19 2D-matrix relabeling and filling in coordinates (Fig. **S33**a), thereby embedding the physics-graph information into the data, with adjacent sites separated by a unit matrix element. Fig. **S33**b and **S33**c illustrate the

Fully-connected deep neural network classifier and Physics-graph-informed CNN classifier.

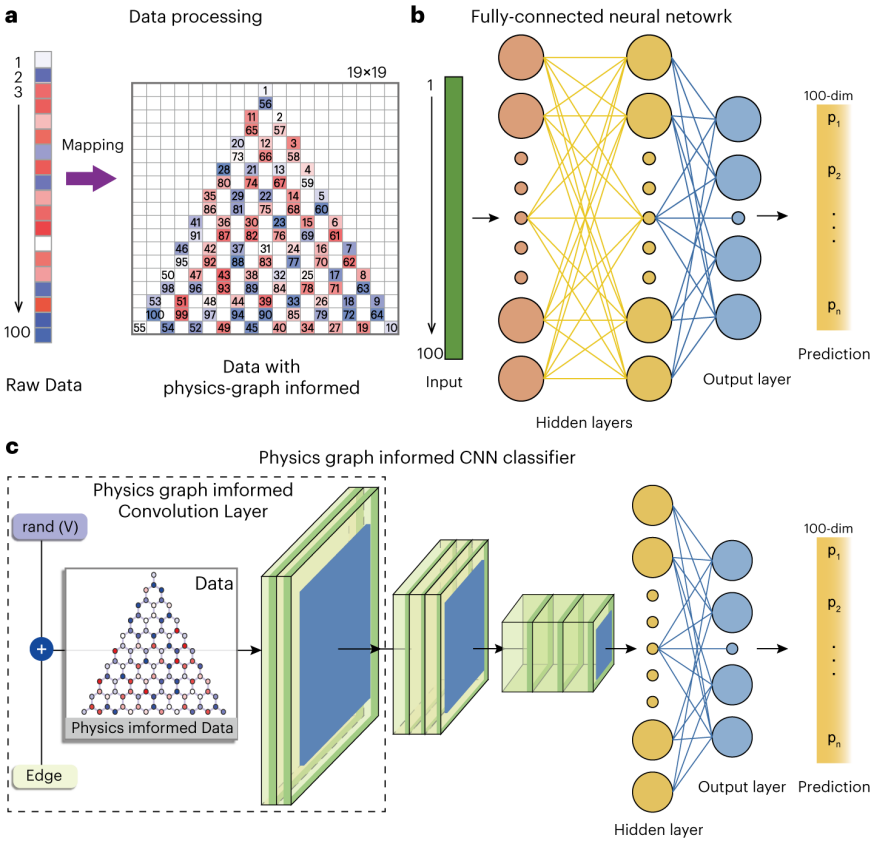

FIG. **S33**. **The comparison of structural data between Fully-connected deep neural networks classifier and physics-graph-informed CNN classifier.** (a) The graph information embedding process from 1×100 1D line-shape data to structural 19×19 2D data. The data is mapped to the corresponding positions with identical labels in a 2D matrix, and the remaining positions are kept as zero. (b) Fully-connected deep neural network classifier with 1D line-shape data input. (c) Physics-graph-informed CNN classifier with structural data input.

Theoretically, increasing the depth of a network is expected to improve performance. However, it will consume more computational resources and increase the time consumption. A suitable network scale can maintain the balance between network performance and resource consumption. The Adaptive Moment Estimation (Adam) optimizer [11] is used to optimize the network parameters, employing a learning rate of 0.001 and a weight decay rate of 0.001. The dropout method [12] is applied to the hidden layers of both FCNN and PGI-CNN to mitigate overfitting, with a dropout probability of $p = 0.5$. The results are shown in Fig. **S34**. Initially, a common CNN framework is introduced by using a pooling

layer after each convolution layer. With a three-layer CNN, the cross-entropy loss is defined as:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{S21}$$

which is decreased to 0.5, and the test accuracy finally increased to approximately 80%. With error correction checks (ECC) mechanism, such accuracy can ensure the data integrity, indicating that the classifier operates effectively within this structure. However, by removing the pooling layer after the first convolution layer while maintaining the subsequent layers, the loss can be as low as 0.15, and the predicted accuracy on arbitrary test data can be up to approximately 95%, which is quite a high value. This improvement is attributed to the first convolution layer functioning as a tight-binding Hamiltonian. A pooling layer after a functioning layer would compromise some physical information. However, the following convolution layer only functions as further feature extraction, so that the pooling layer can reduce the resource consumption while keeping high accuracy.

We also increase the network scale to four-layer PGI-CNN and five-layer PGI-CNN, the results are shown in Fig. **S34**c and **S34**d. It can be observed that the accuracy of test data is slightly higher than 95%. It means that a three-layer PGI-CNN can achieve the balance between performance and resource consumption. In our work, a three-layer PGI-CNN is utilized.

D. The performance comparison between normal neural network architectures and PGI-CNN

Making the network deeper can increase the accuracy, but will multiply the computing and memory costs. The current network balances accuracy and computation cost. It should be emphasized that good accuracy derives from the physics-graph-informed mapping between the graph structure hardware and the neural network architecture. For comparison, we also construct a full-connected neural network (FCNN) with more neurons for the same dataset, and a common convolutional neural network (CNN) to deal with the data. Results are shown in Fig. **S35** and **S36**. It can be observed that the network with one hidden layer (total of three layers) and dropout probability of 0.5 to avoid over-fitting can achieve an accuracy of approximately 65% on test data (Fig. **S35**a). Such accuracy makes it difficult for the ECC check algorithm to completely restore the correct information. However, by increasing

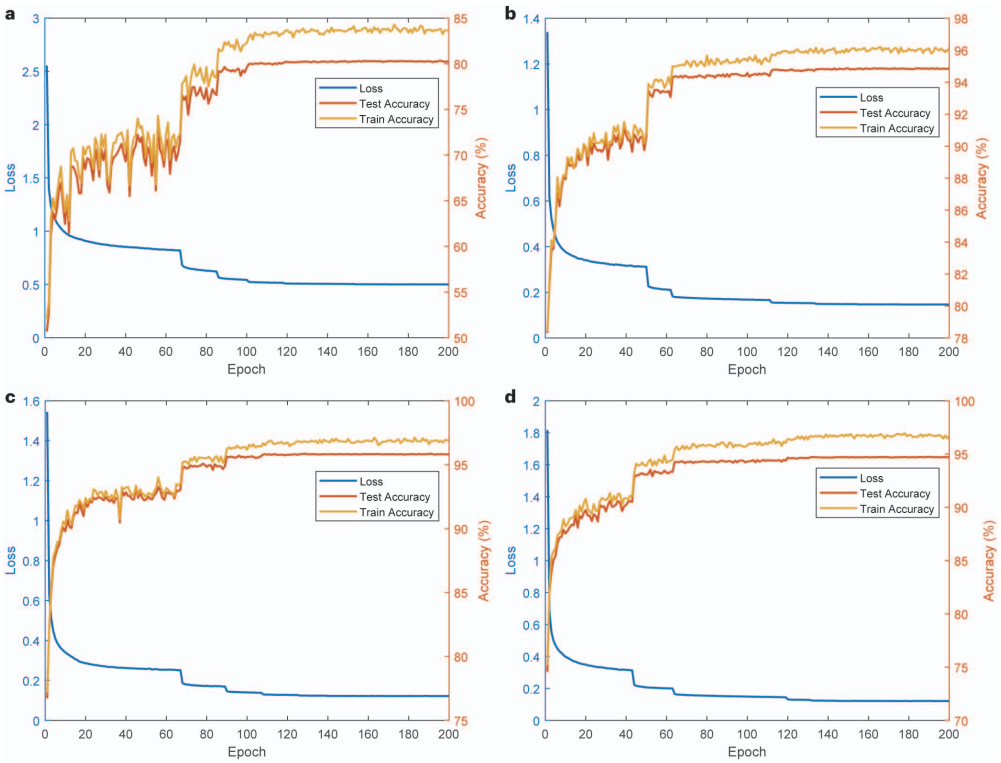

FIG. **S34**. **Training result and network optimization trial of PGI-CNN classifier.** (a) 3-CNN layers with max pooling layer after first convolution layer. (b) 3-CNN layers without a max pooling layer after the first convolution layer, which could extract physics graph information (c) 4-CNN layers with a max pooling layer after the first convolution layer. (d) 5-CNN layers with a max pooling layer after the first convolution layer.

the hidden layers to two layers, the accuracy of test data is even decreased to 45% (Fig. **S35**b). The dataset may not be sufficient to support the network, and there is over-fitting that further degrades the performance. Figures **S35**c and **S35**d show the FCNN with one hidden layer and two hidden layers, but without a dropout mechanism. It can be observed that the over-fitting is more serious, so that the accuracy on the training dataset can be much higher due to the strong fitting capability of FCNN. The accuracy of the test data set is so poor that even the ECC check algorithm cannot ensure the data integrity when it is used for an information encryption application. Moreover, if we reshape the data to a 10×10 two-dimensional matrix, and use a convolution neural network to deal with it, the performance becomes even worse than FCNN, and the accuracy decreases from 65% to less than 45%. It is because the mechanism drops some connections among different sites, which

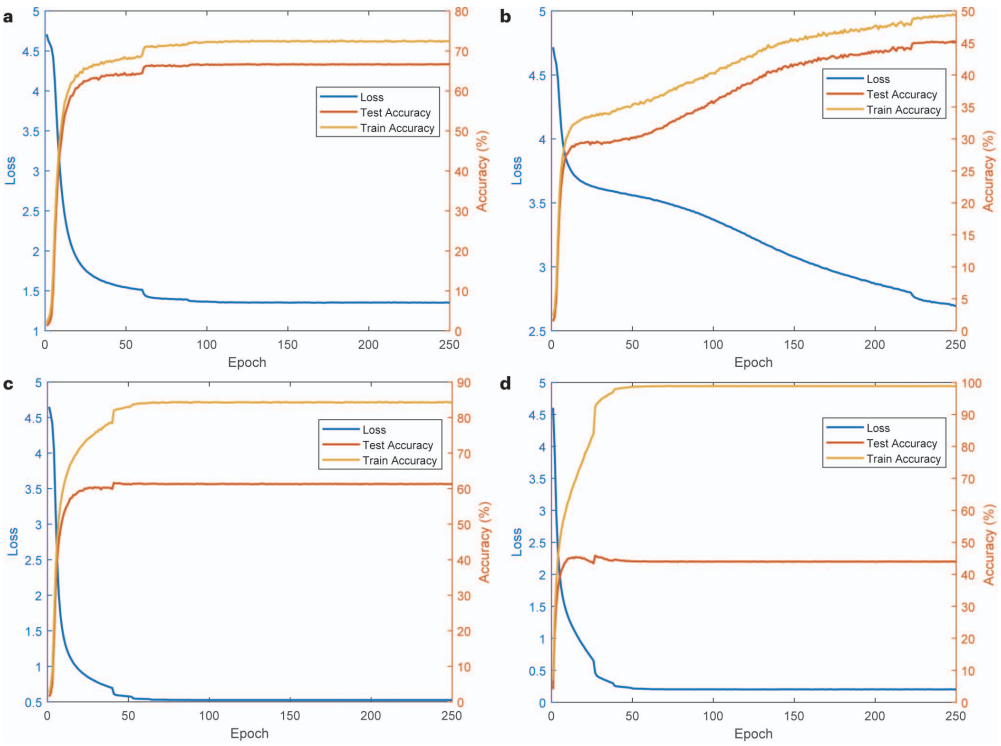

FIG. **S35**. **Training result and network optimization trial of fully connected neural network.** (a) 3-layer network (1-hidden layer with dropout layer (p=0.5)). (b) 4-layer network (2-hidden layer with dropout layer (p=0.5)). (c) 3-layer network (1-hidden layer without a dropout layer). (c) 4-layer network (2-hidden layer without dropout layer).

further breaks the data correlation. It is obvious that without the participation of physics features, it can hardly obtain the effective feature extraction capability by relying only on pure common neural network learning due to the random disorder-induced localization.

E. Deep-learning model-driven localization characteristic analysis

With the help of the PGI-CNN classifier, we can fast label the random disorder data that can induce strong localization to find out their localized position, so that the characteristic of Anderson localization on such systems can be analyzed through statistical results.

A large number of random numbers with different seeds are sampled to evaluate the localization effect. Each time, the localized central positions are recorded to make the statistics. We provide the statistical results for relatively smaller and larger disorder strengths. When the disorder strength $S = 5$, the disordered strength of on-site terms is in the range of [-2.5,

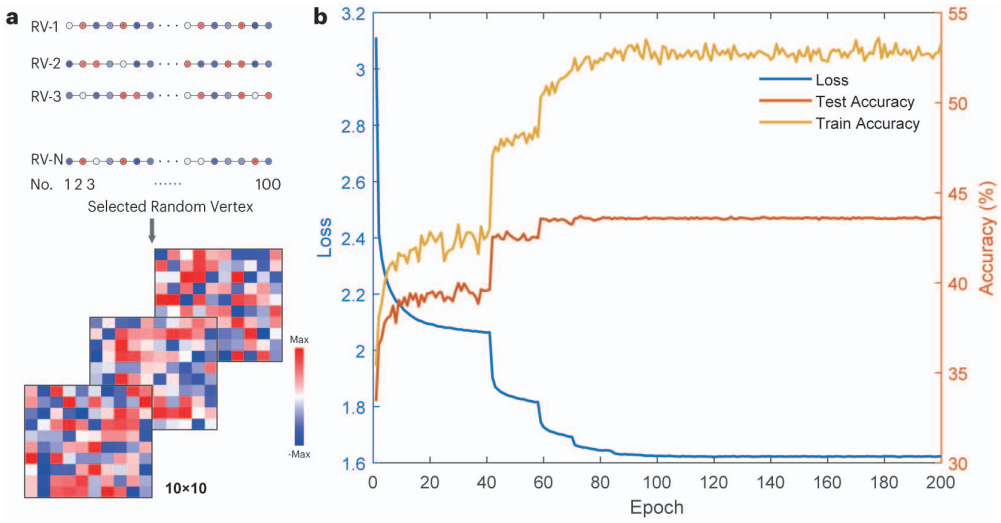

FIG. **S36**. **Training result with simple reshaping to a 2D matrix and using a conventional CNN for classification.** (a)Data reshaping procedure. (b)Training process and results.

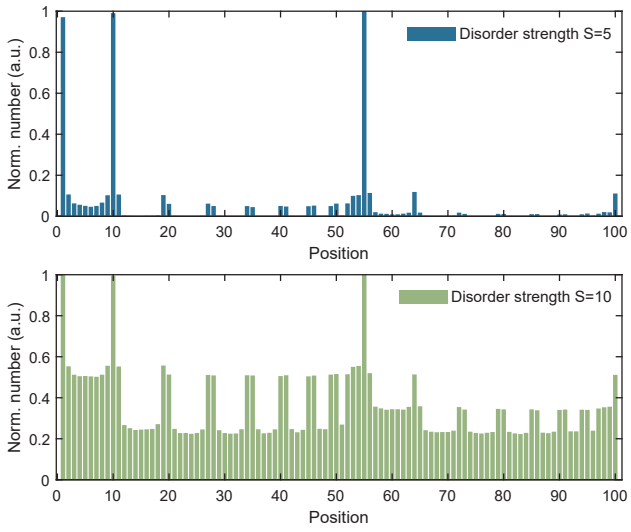

FIG. **S37**. **Probability of the localization position with disorder strength S=5 and S=10.**

2.5]. It can be observed that the localized position is almost at sites 1, 10, and 55, which indicates the three vertices at the three corners of the triangle geometry. Such positions are the corner of topology, which intrinsically promotes localization. For a weaker disorder strength, only the three positions can have localization, and their localized probability is approximately identical if the number of statistics is large enough. By increasing the disorder strength, the position capable of effective localization is also increased. When $S = 10$,

it can be observed that all positions of the system can have localization, shown in Fig. **S37**.

F. The Physics-graph-informed diffusion model (PGI-diffusion)

This diffusion model is a conditional generative model based on a U-Net backbone, designed to generate target data distributions by progressively denoising inputs through a sequence of reversible transformations [13]. When designing the network basis, physics-graph information, including the topological connection graph and tight-binding rules, is embedded. The network structure is listed below.

The PGI-diffusion Network Architecture

```
ResBlock(channels, cond_dim):
    Input: feature map x, condition vector cond
    h = Mish( GroupNorm( Conv2d(x, channels, kernel=3, padding=1) ) )
    cond_proj = Linear(cond, output_dim=channels)
    cond_proj = reshape(cond_proj)  // shape to [B, channels, 1, 1]
    h = h + cond_proj
    h = Mish( GroupNorm( Conv2d(h, channels, kernel=3, padding=1) ) )
    Output: x + h

UNet(in_channels=2, out_channels=1, base_channels, cond_dim, label_dim,
   Filter):
    Input: x, time embedding t_emb, labels
    cond_time  = Linear(t_emb)
    cond_label = (labels provided) Embedding+Linear(labels)
    cond = cond_time + cond_label


    //--- Encoder (Downsampling) ---
    x1 = Mish( Conv2d(x, out_channels = base_channels,
                      kernel=3, padding=1) )
    x1 = ResBlock(base_channels, cond_dim)(x1, cond)
```

```
21
22      x2 = Mish( Conv2d(x1, out_channels = base_channels*2,
23                        kernel=3, stride=2, padding=1) )
24      x2 = ResBlock(base_channels*2, cond_dim)(x2, cond)
25
26      x3 = Mish( Conv2d(x2, out_channels = base_channels*4,
27                        kernel=3, stride=2, padding=1) )
28      x3 = ResBlock(base_channels*4, cond_dim)(x3, cond)
29
30      x4 = Mish( Conv2d(x3, out_channels = base_channels*8,
31                        kernel=3, stride=2, padding=1) )
32      x4 = ResBlock(base_channels*8, cond_dim)(x4, cond)
33
34      //--- Decoder (Upsampling) ---
35      x4_up = Upsample(x4, target_size = size(x3))
36      x5 = Concatenate(x4_up, x3)
37      x5 = Mish( Conv2d(x5, out_channels = base_channels*4, kernel=3,
            padding=1) )
38      x5 = ResBlock(base_channels*4, cond_dim)(x5, cond)
39
40      x5_up = Upsample(x5, target_size = size(x2))
41      x6 = Concatenate(x5_up, x2)
42      x6 = Mish( Conv2d(x6, out_channels = base_channels*2, kernel=3,
            padding=1) )
43      x6 = ResBlock(base_channels*2, cond_dim)(x6, cond)
44
45      x6_up = Upsample(x6, target_size = size(x1))
46      x7 = Concatenate(x6_up, x1)
47      x7 = Mish( Conv2d(x7, out_channels = base_channels, kernel=3, padding
            =1) )
```

```
    x7 = ResBlock(base_channels, cond_dim)(x7, cond)

    out = Conv2d(x7, out_channels = out_channels, kernel=1)
```

The core training process and inference process include:

Forward Diffusion Process. The forward process gradually transforms a clean data sample $x_0 \in \mathbb{R}^D$ into a noisy version x_t over $t = 1, \ldots, T$ steps by incrementally adding Gaussian noise. At any given time step t, the noisy sample is generated as:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{S22}$$

where the cumulative noise scaling factor is defined by:

$$\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s, \quad \text{with} \quad \alpha_s = 1 - \beta_s. \tag{S23}$$

Here, β_s denotes a predefined noise schedule. At $t = 0$, we have $\bar{\alpha}_0 = 1$ and no noise is added, while for large t the sample x_t becomes increasingly dominated by noise.

v-prediction Parameterization. Instead of directly predicting the noise ϵ, we adopt the v-prediction formulation. In this setup, an auxiliary variable v is defined as [14]:

$$v = \sqrt{\bar{\alpha}_t}\, \epsilon - \sqrt{1 - \bar{\alpha}_t}\, x_0. \tag{S24}$$

The model is then trained to predict v from the noisy observation x_t and the time step t. This reparameterization has been found to yield improved training stability as well as enhanced sample quality.

Training Objective. The model is optimized by minimizing the mean squared error between the true v, derived from Equation (S24), and the network prediction $v_\theta(x_t, t)$. The loss function is given by:

$$L = \mathbb{E}_{t, x_0, \epsilon} \left[\| v - v_\theta(x_t, t) \|^1 \right]. \tag{S25}$$

This objective directly encourages accurate predictions of v, which in turn facilitates effective reconstruction of the clean data.

Classifier-Free Guidance. For conditional generation tasks, we employ classifier-free guidance to enhance sample fidelity [15]. During training, the conditioning label y is randomly dropped with a predefined probability (i.e., replaced with a null token, denoted by

$\varnothing$). Consequently, the neural network learns both conditional $v_\theta(x_t, t, y)$ and unconditional $v_\theta(x_t, t, \varnothing)$ predictions. At inference time, the guided prediction is computed as:

$$\tilde{v}_\theta(x_t, t, y) = v_\theta(x_t, t, \varnothing) + \gamma \left[v_\theta(x_t, t, y) - v_\theta(x_t, t, \varnothing) \right], \tag{S26}$$

where γ (often greater than 1) is the guidance scale that controls the strength of the conditioning. This formulation effectively steers the sampling process towards the desired conditional distribution while retaining robustness from the unconditional model.

Data Reconstruction and Reverse Diffusion. Once a prediction $v_\theta(x_t, t)$ is available, the clean data x_0 can be reconstructed from x_t using the inversion formula:

$$x_0 = \sqrt{\bar{\alpha}_t}\, x_t - \sqrt{1 - \bar{\alpha}_t}\, v. \tag{S27}$$

The reverse (denoising) process is modeled as a Markov chain of Gaussian transitions. Specifically, the transition probability from step t to $t-1$ is given by:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\Big(x_{t-1};\, \mu_\theta(x_t, t),\, \sigma_t^2 I\Big), \tag{S28}$$

where the neural network outputs both the mean $\mu_\theta(x_t, t)$ and the variance σ_t^2. These parameters guide the iterative denoising from a state of nearly pure noise back to a high-fidelity x_0.

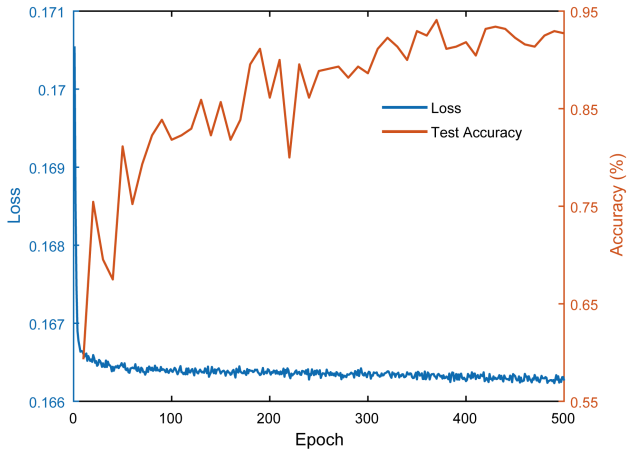

FIG. **S38**. **The training loss with PGI-DDPM training process.**

Figure **S39** illustrates data produced by the fully trained PGI-DDPM model. For visualization, we selected 11 labels and randomly generated 10 samples for each. The resulting distributions are remarkably diverse, showing that the mapping from labels to samples is far from a simple one-to-one correspondence.

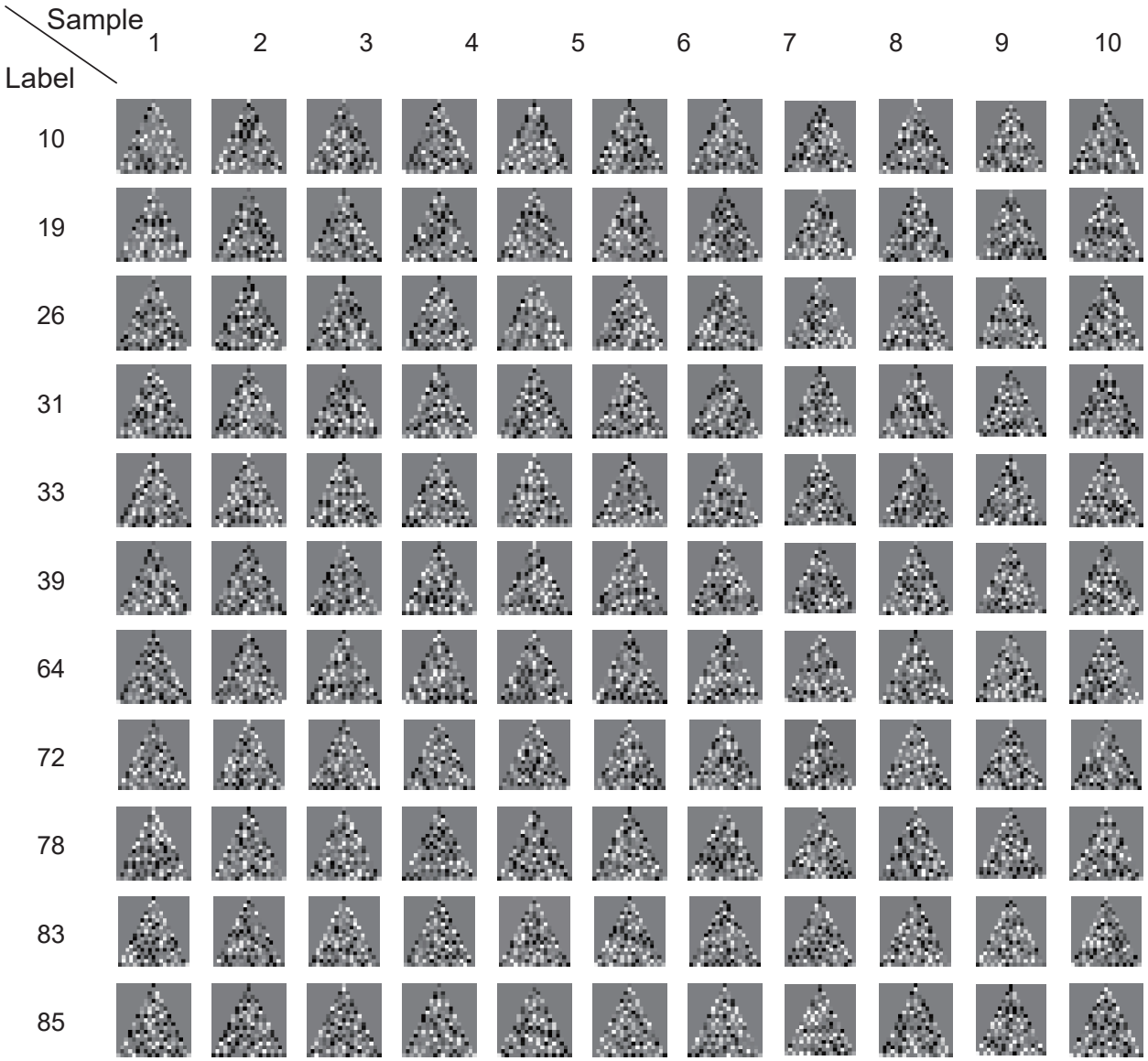

FIG. **S39**. **Image-visualization of generated data by PGI-diffusion model.** Eleven different localization position labels are randomly selected, and in each label, ten different samples are demonstrated.

G. Generative capabilities of the PGI-diffusion model and data analysis

Through training, the PGI-Diffusion model effectively acquires the salient features required to capture Anderson localization in complex physical systems. Leveraging supervision from labeled data, it further learns to map and delineate the regions of the high-dimensional parameter space occupied by the disorder-strength vectors corresponding to each localized position in the physical model. At generation process, the model takes two inputs: the

label of the target localized position and a random noise vector with the same shape as the disorder strength vector, as shown in Fig. **S40**. Using the embedding of this random noise vector as an index, the model samples within the high-dimensional parameter space to continuously generate disorder strength vectors that satisfy the given label constraint. The relationship between the localized position and the generated results is "one-to-many". However, for any given input pair of the required localized position and random vector, the model's output is "one-to-one" intrinsically. The arbitrarily chosen, unconstrained random noise vectors serve to control and diversify the generated samples.

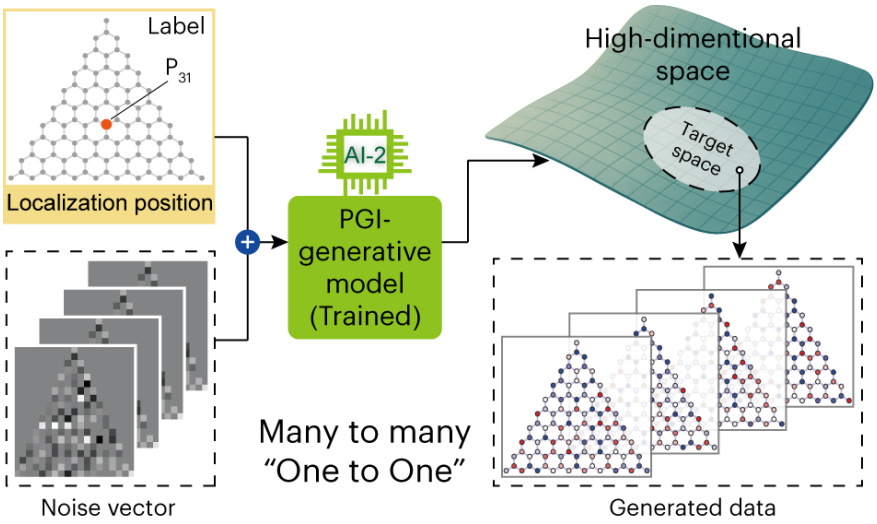

FIG. **S40**. **The explanation of the "one-to-many" capability of generative models that generate multiple disorder strength vectors from a single label.** The model takes two inputs: the label of the target localized position and a random noise vector with the same shape as the disorder strength vector. Using the embedding of this random noise vector as an index, the model samples within the high-dimensional parameter space to continuously generate disorder strength vectors that satisfy the given label constraint. The arbitrarily chosen, unconstrained random noise vectors serve to control and diversify the generated samples.

In the following, we present evidence that the data produced by this AI model are not simply training samples output by using a random seed to sample from the dataset as a one-to-many mechanism. Instead, they are genuinely generated in accordance with the system's physical principles (namely, precisely realizing controlled Anderson localization) and are entirely new parameter vectors different from those seen during training—truly achieving

exploratory generation over the vast entropy space.

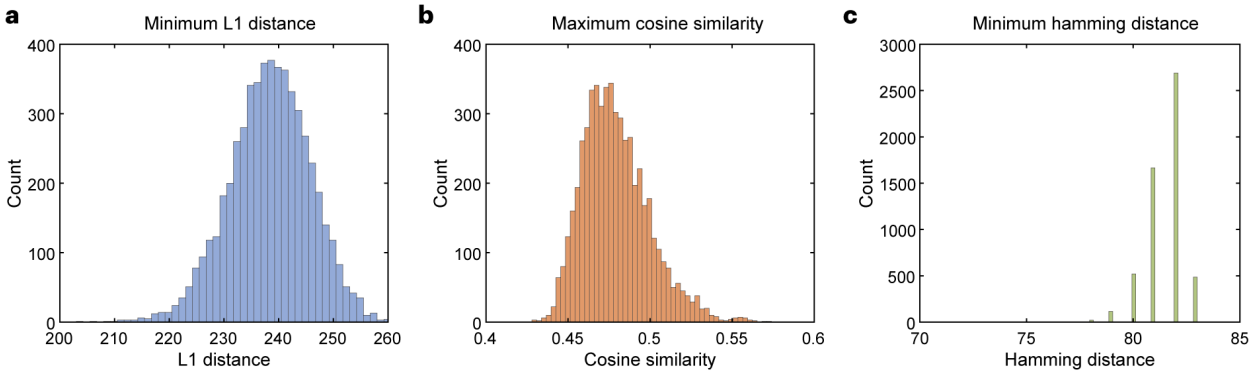

FIG. **S41**. **Statistical validation results for the similarity between the model-generated disorder strength data and the data in all training datasets.** 5500 samples are uniformly generated for all localization positions. (a) The statistical results of L1 distance. The value falls within 200–260, with a minimum of 203, which means that each generated state is at least 203 away from the test set (given that we have 100 points, the average per-point distance exceeds 2). (b) The statistical results of cosine similarity. The value ranges from 0.428 to 0.573, indicating that the generated samples differ from those in the dataset. (c) The statistical results of hamming distance. The value is distributed between 74 and 83, indicating that there are at least 74 points whose values differ by more than 0.5 between the generated dataset and the existing data.

We analyzed 5,500 effectively localized generated samples. For each sample, we adopt three representative metrics for measuring the similarity between generated sample and all 720,000 samples in training dataset, including L1 distance (the sum of absolute differences across components between two vectors in a high-dimensional space; smaller values indicate greater similarity), cosine similarity (the degree of directional or distributional-shape similarity between two vectors; larger values indicate greater similarity), and Hamming distance (the number of positions at which the corresponding components differ; smaller values indicate greater similarity). We compare the generated data with all the training data to assess their similarity.

Both the generated data and the training set lie within the range [-5, 5]. The statistical results are shown in Fig. **S41**. The L1 distances fall within 200–260, with a minimum of 203, which means that each generated state is at least 203 away from the test set (given that we have 100 points, the average per-point distance exceeds 2). The cosine similarity ranges

from 0.428 to 0.573, indicating that the generated samples differ from those in the existing dataset. The Hamming distance is distributed between 74 and 83, indicating that there are at least 74 points whose values differ by more than 0.5 between the generated dataset and the existing data. These results show that our model produces entirely new states that are distinct from the existing data; as verified by the tight-binding model, they still realize Anderson localization at the prescribed position. This indicates that the model's generative domain has expanded into a vast entropy space.

Importantly, this model is not confined to reproducing disorder patterns from the training dataset. It can generate physically valid, previously unseen configurations that still satisfy the required localization properties.

H. The cVAE-based generative network design and optimization

A conditional Variational Autoencoder (cVAE) is a type of generative model that can generate specific and controlled data outputs [16]. The key point of a cVAE is its ability to incorporate conditional information into decoder networks, enabling it to learn a more nuanced representation of the data. One of the primary advantages of cVAEs is their ability to generate data that adheres closely to desired conditions, making them useful in applications where specific attributes are required.

Similar to the PGI-diffusion model, we also introduce the PGI mechanism into the classical cVAE to construct a PGI-cVAE network and use it for the controllable Anderson Localization generation. As Fig. **S42** shows, it incorporates a PGI-CNN for encoding and a standard CNN for decoding. The encoder processes all input data and maps them in a latent space, represented by a distribution $N(\mu, \sigma)$ and characterized by the mean (μ) and variance (σ) of the distribution. During the generation procedure, a sample vector $(z_1, ...z_n)$ is randomly drawn from this distribution and concatenated with the embedded label condition $(c_1, ...c_n)$. This composite vector is then input into the decoder to reconstruct the output data rand(V). With varying network weights and output layers, the encoder is utilized to transform the disorder parameter space to a Gaussian-like distribution. For the decoder design, sampled data from the latent space is used in conjunction with the embedded labels as network input. The deconvolution layer and leaky-ReLU activation function [17] are utilized for the sampling process. This reconstruction is influenced by the probabilistic nature of

the sampling process, which enables the model to generate diverse outputs while adhering to the specified conditions. To build the loss function, meaning absolute error (MAE) (Eq. S30) and Kullback-Leibler divergence (KLD) (Eq. S29) work synergistically to regularize both the error of each disorder vector and the data distribution.

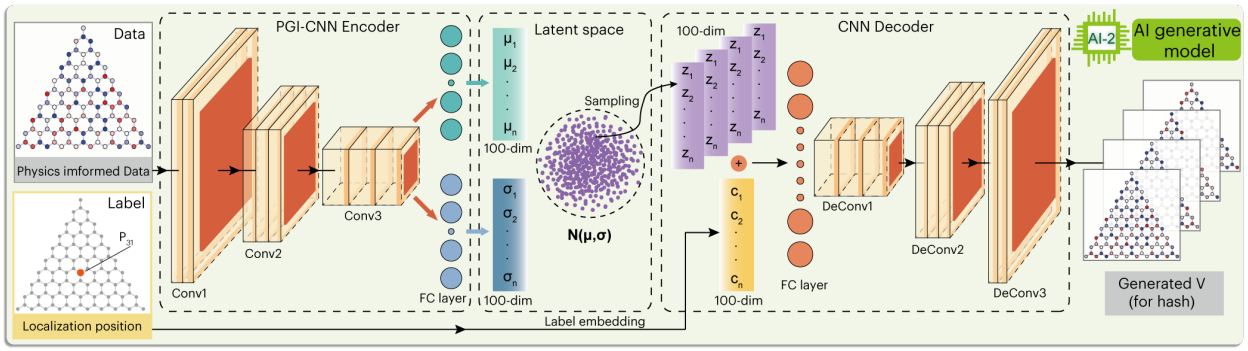

FIG. **S42**. **The cVAE-based generative network architecture.**

The cVAE Network Architecture

```
Encoder:
    Convolution(in_channel=1, out_channel=256, kernel=3x3, stride=1,
        padding=1)
    BatchNorm
    LeakyReLU
    Convolution(in_channel=256, out_channel=512, kernel=3x3, stride=1,
        padding=1)
    BatchNorm
    LeakyReLU
    MaxPooling
    Convolution(in_channel=512, out_channel=1024, kernel=3x3, stride=1,
        padding=1)
    BatchNorm
    LeakyReLU
    MaxPooling
    Linear(input=36864, output=1024)
    BatchNorm
```

```
    LeakyReLU

Latent Space:
    fc_mu: Linear(input=1024, output=100)
    fc_logvar: Linear(input=1024, output=100)

Decoder:
    Linear(input=200, output=1024)
    BatchNorm1d
    LeakyReLU
    Linear(input=1024, output=36864)
    BatchNorm1d
    LeakyReLU
    Deconvolution(in_channel=1024, out_channel=512, kernel=3x3, stride=2,
        padding=1)
    BatchNorm
    LeakyReLU
    Deconvolution(in_channel=512, out_channel=256, kernel=3x3, stride=2,
        padding=1)
    BatchNorm
    LeakyReLU
    Deconvolution(in_channel=256, out_channel=1, kernel=3x3, stride=1,
        padding=2)
```

For comparison, we have listed different loss functions to regularize numerical error, including the mean squared error (MSE) (Eq. S31), mean absolute error (MAE) (Eq. S30), together with the KLD as the overall loss function. The Adam optimizer is employed for optimization, with a learning rate of 0.001 and no weight decay. The cVAE network did not utilize the dropout method. The training results are shown in Fig. **S43**. Based on the result, we select the MAE loss to construct the final loss function.

$$\mathrm{KL}[q(z|x)\|p(z)] = -\frac{1}{2}\sum_{i=1}^{d}\left(1+\log(\sigma_i^2)-\mu_i^2-\sigma_i^2\right) \tag{S29}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i| \tag{S30}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \tag{S31}$$

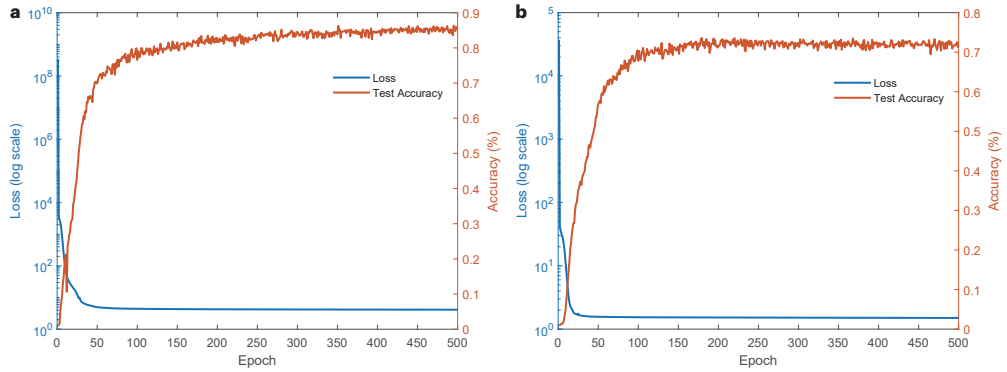

FIG. **S43**. **The training result with different loss functions.** (a) MAE loss. (b) MSE loss.

To visualize the training and test process, we reshape the generated data by the cVAE in each epoch in Fig. **S44**. Compared with the original disorder data, in the initial several epochs, the data at each site have small disorder characteristics, after 20 epochs, their distribution is similar to the original data, and at the final epoch, their characteristics are almost the same as the original data.

As a result, we show the complete generated disorder result for all 100 labels. The correctness of labels is examined by TBM. As shown in Fig. **S45** demonstrated in 2.5D and 2D versions. From Fig. **S45**a the unexpected data that localized to other positions are shown, which is trivial. From Fig. **S45**b, the correctness of all sites is higher than 81%, which is high enough for information cryptography.

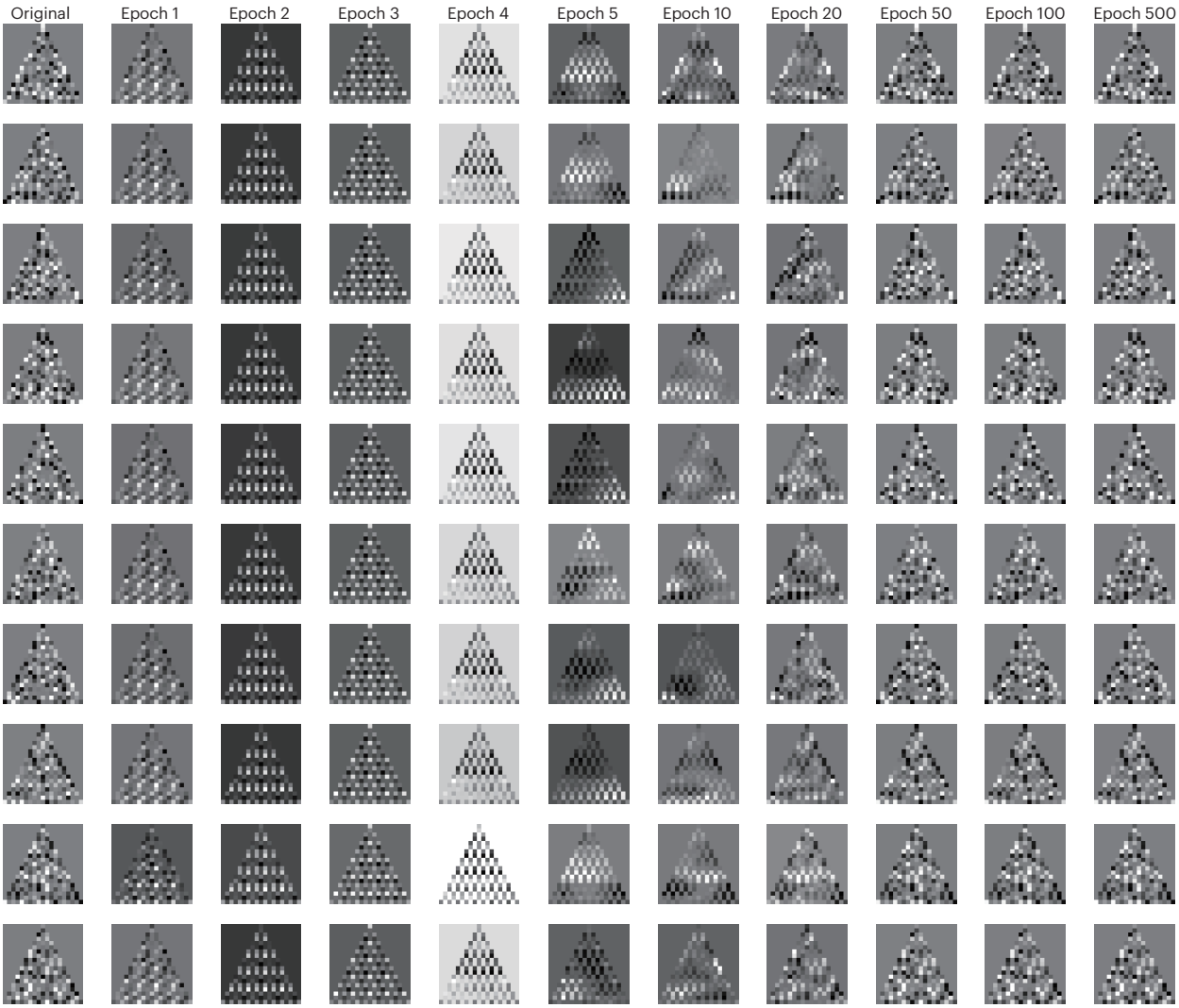

FIG. **S44**. **Image of the network's output data during the training process.** It can be observed that with the epoch increasing, the distributions are rapidly shifted from uniform to the target feature.

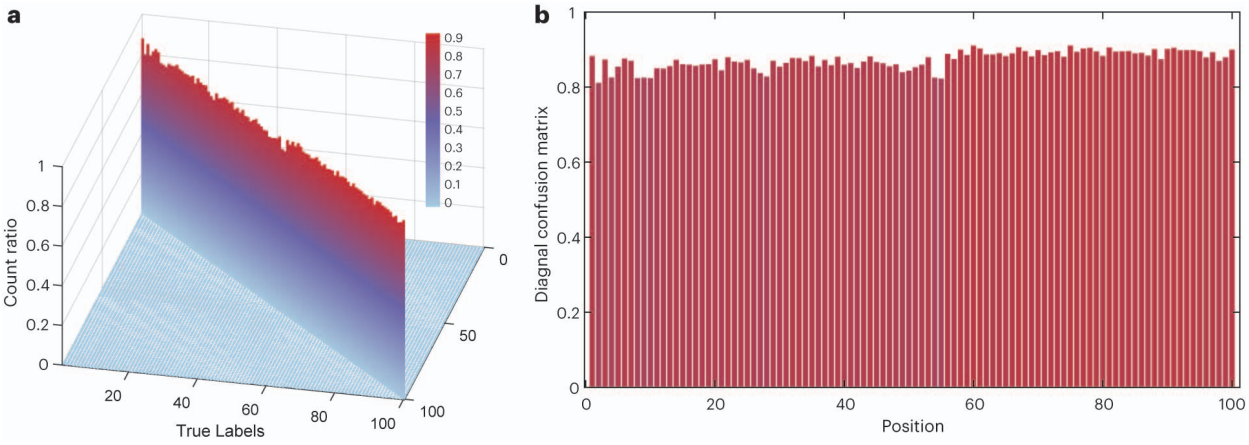

FIG. **S45**. **The confusion matrix of the generated result.** (a) Complete confusion matrix. (b) The correctness of each label.

I. More data results of arbitrary position localization

In this section, we show the theoretical and experimental results with different disorder strengths. When the disorder strength is increased, energy can be localized from certain boundaries that are easy to localize, to arbitrary positions in the whole PTC.

With relatively small disorder strength (the onsite disorder strength $S = 5$), localization can only happen on three corner points, as shown in Fig. **S46**. By increasing the disorder strength to $S = 8$, it can easily localize to all corner points and edge points, but the inner bulk position needs precise screening, as depicted in Fig. **S47**. By increasing the disorder strength to $S = 10$, it can easily localize to arbitrary positions including corner points, edge points, and bulk points, as depicted in Fig. **S48**.

However, the distribution of the remaining small amount of energy deviates from the simulated results. Because the remaining energy functions like a higher-order small quantity, which is very sensitive to the parasite parameters of the circuit components. The simulation can only give the parameter fluctuation at the statistical level, but the unique fluctuation for each component in a real experiment. By overlapping a series of wave functions for certain localized positions and taking the average, only the central position will be enhanced, which has high consistency between simulation and experiment. However, due to the generated vector leading to a high localization rate, once a threshold is determined, the central position for a single time is very stable no matter in simulation or experiment. Such a phenomenon inspires us to a unique physics-informed hash-based probabilistic information encryption method, in which the clear local points can function as the coordinates for the information space.

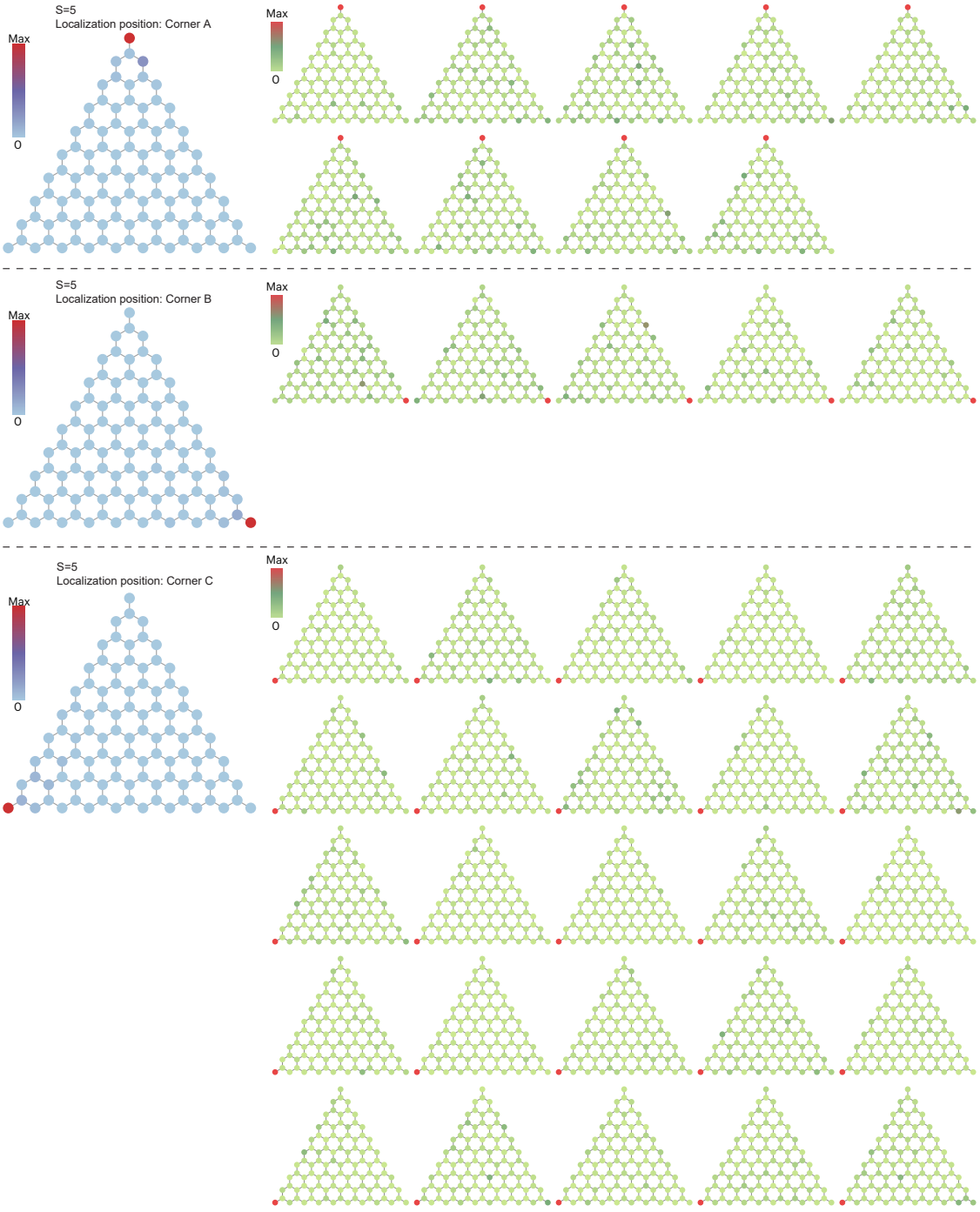

FIG. **S46**. **The simulated and experimental Anderson localization by different disorder parameters with disorder strength S = 5.**The localization can only happen on three corner points.

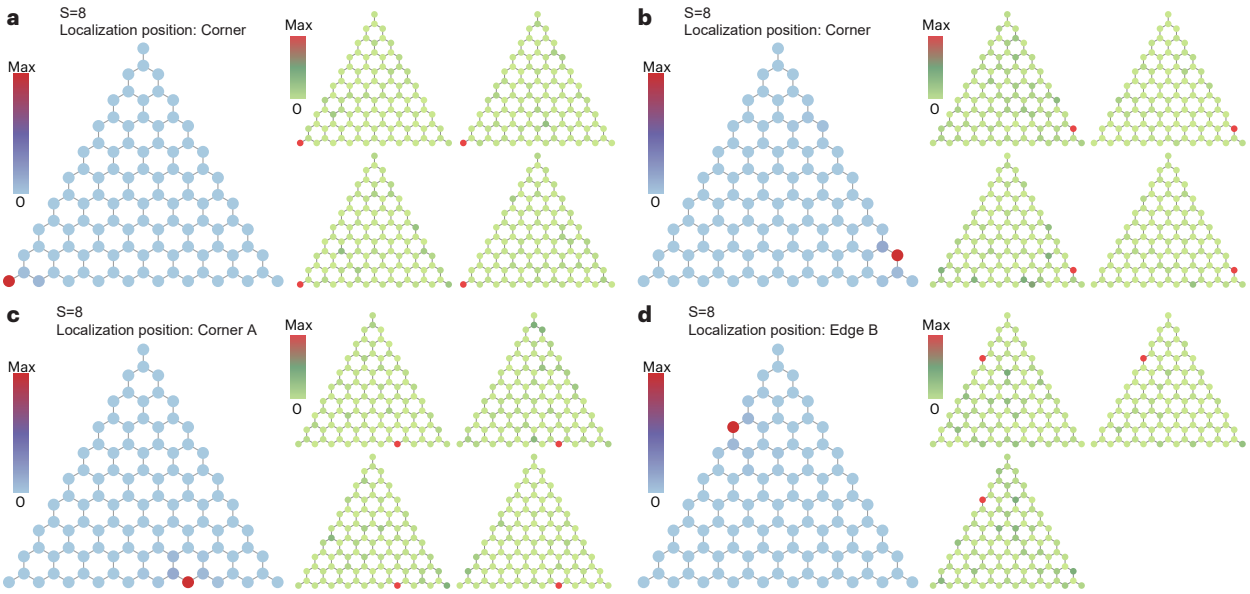

FIG. **S47**. **The simulated and experimental Anderson localization by different disorder parameters with disorder strength S = 8.**

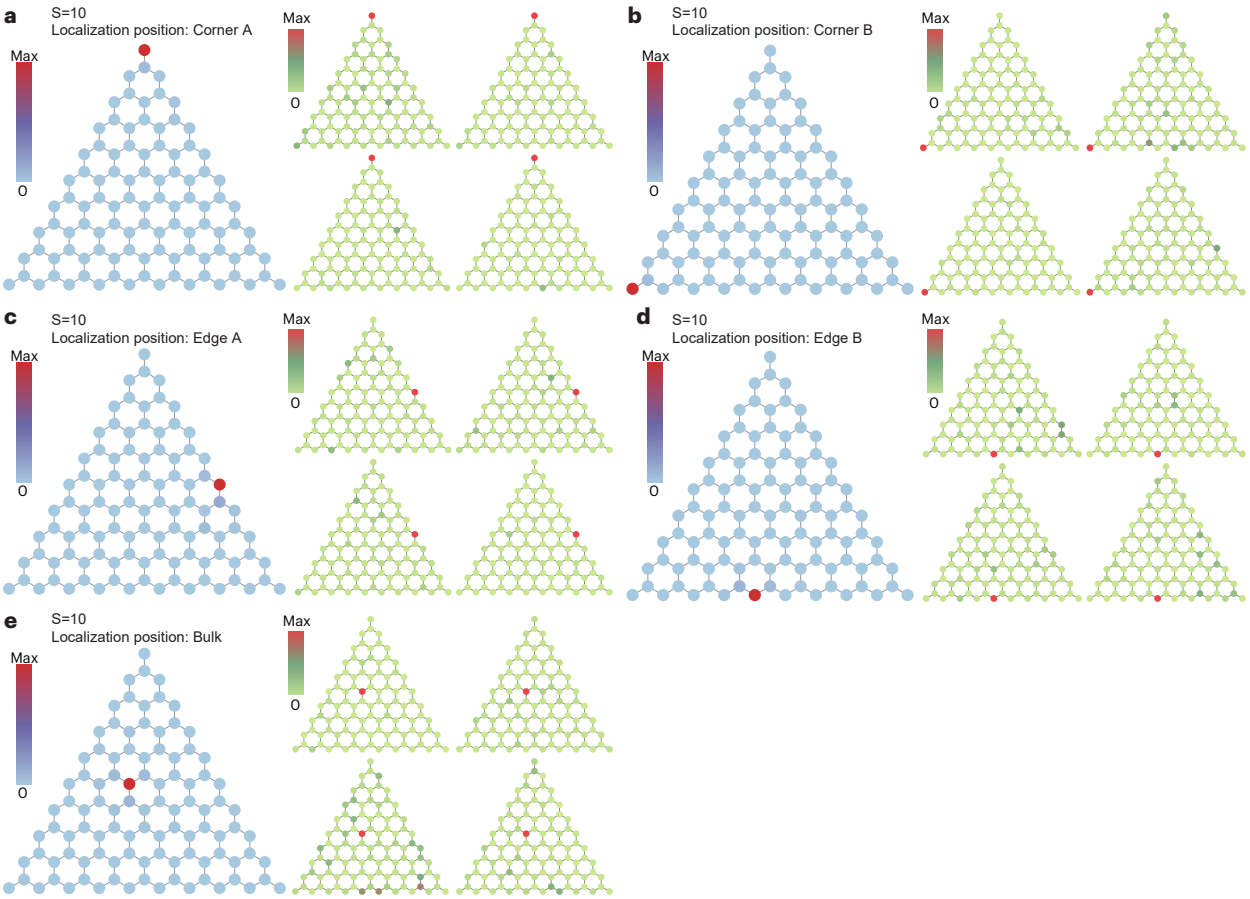

FIG. **S48**. **The simulated and experimental Anderson localization by different disorder parameters with disorder strength S = 10.**

S5. ANDERSON HASH-BASED PROBABILISTIC INFORMATION ENCRYPTION USING CONTROLLABLE ANDERSON LOCALIZATION

A. Anderson localization analysis with all disorder degree of freedom

Only on-site disorder: [rand(V), E]

For only on-site disorder, at low disorder strength, the energy is hardly focused, and the average IPR approaches zero. From the wave function, it can be observed that energy is distributed in almost all sites. With the increase of disorder strength, the maximum IPR and average IPR increase rapidly. When disorder strength is larger than 10, the maximum IPR approximately approaches 1, demonstrating that in this region, parameters for strong enough localization can be found. However, the average IPR is about 0.4, which means that there still exist many alternatives with relatively low localization. To increase the optimization efficiency, consciously increase the disorder strength, and the average IPR gradually increases, but the growth rate tends to flatten. It means that after the disorder strength is over 40, it will not bring further advantages for localization. At low disorder strength, energy cannot be focused on any site. By increasing the disorder strength, the three corners first get strong localization, then the edge region, and the bulk. With strong disorder strength, almost all sites can easily get the localized solution.

Only off-site disorder: [V, rand**(E)]**

In the modulation type $G = [V, \mathrm{rand}(E)]$, i.e., only the off-site hopping is randomly disordered. In this case, we keep the on-site potentials to zero. Due to the on-site terms being identical, there is a series of degenerate eigenmodes that exist with zero eigenenergy. It can be observed that increasing the disorder strength can enlarge the fluctuation range of eigenenergy, but the modes with non-zero eigenenergy all have low IPR. All modes with high IPR are in the zero-energy region. It means that such Anderson localization will concentrate on multiple sites simultaneously, as Fig. **S49** shows. This phenomenon makes one random vector mapped to multiple uncertainty labels. Such "multiple to multiple" mapping violates the Hash definition, thus making it difficult for encryption applications. Moreover, energy focusing on multiple points will lower the intensity of each localized point, which will further decrease the signal-to-noise (SNR) of the information. Fig. **S49**a is the statistical results by taking the average quantity of eigenmodes of $[V, \mathrm{rand}(E)]$, it can be observed that the three corner points are still the easiest to localize. The actual modes for different [V, rand(E)] are

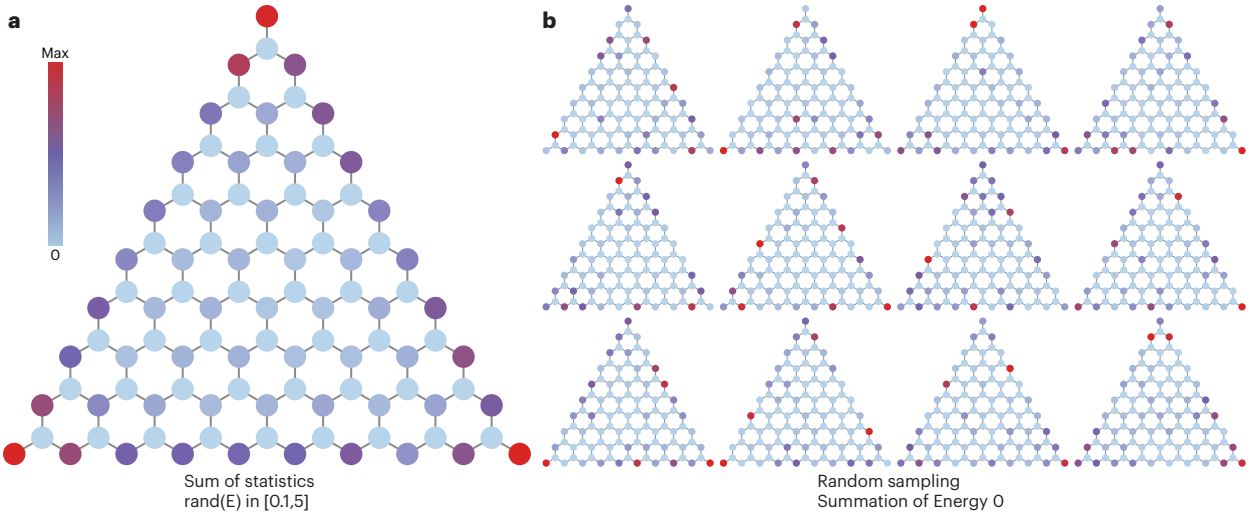

FIG. **S49**. **Mode distribution with multiple points localization.** (a) Summation of statistical results. (b) random sampling, summation of E=0.

exemplified in Fig. **S49**b, which illustrates that the energy is randomly localized to multiple points of the system, and the localized positions are uncontrollable.

B. Additional simulation and experimental data for encryption

Utilizing the programmable circuit system, we simulate and experimentally verify the framework above. In the message figure shown in Fig. **S50**, the pixels labeled in the green region are experimentally demonstrated, and the pixels labeled in the blue region are simulated. As illustrated, 1×100 random vectors are generated for transmitting, the information is loaded onto the hardware, and the measured results show that the information is precisely extracted through the system. We have completely verified all points with the y-axis at disorder strengths S = 8, 9, and 10 by experiment, and all remaining points by simulation, as shown in Figs. **S51**-**S57** (To clarify, for simulation, only part of the data points on rows with disorder strength S = 11, 13, and 19 are shown). Both simulation and experiment show high consistency and fidelity. All the demonstrations illustrate that this framework is a feasible way to encrypt and transmit information accurately.

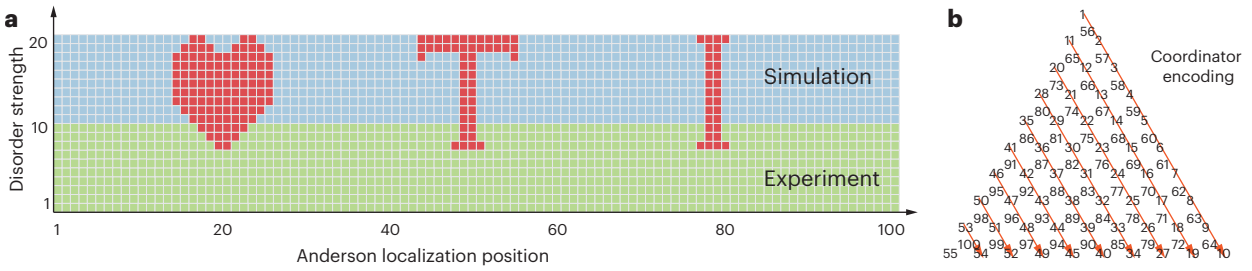

FIG. **S50**. **The whole message by experiment and simulation.** (a) The experimental and simulation verification results. (b) The position labels for coordinate encoding.

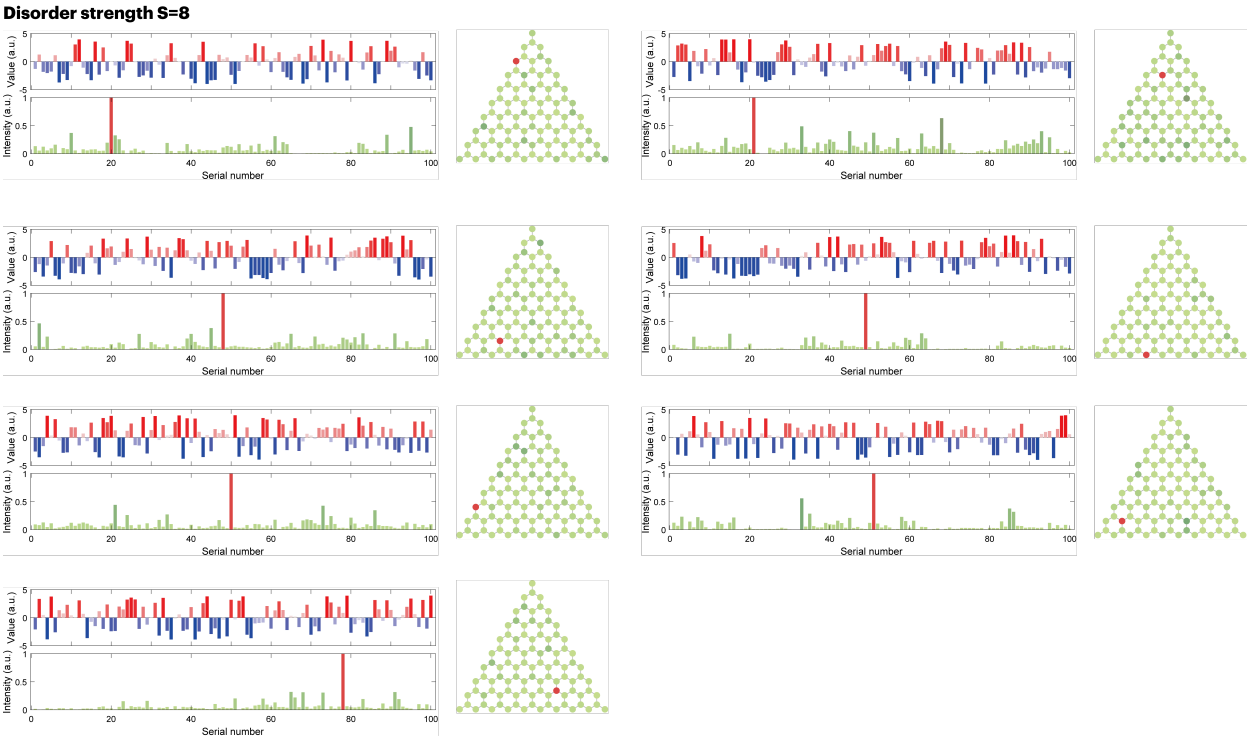

FIG. **S51. Experiment localization points with disorder strength S = 8.**

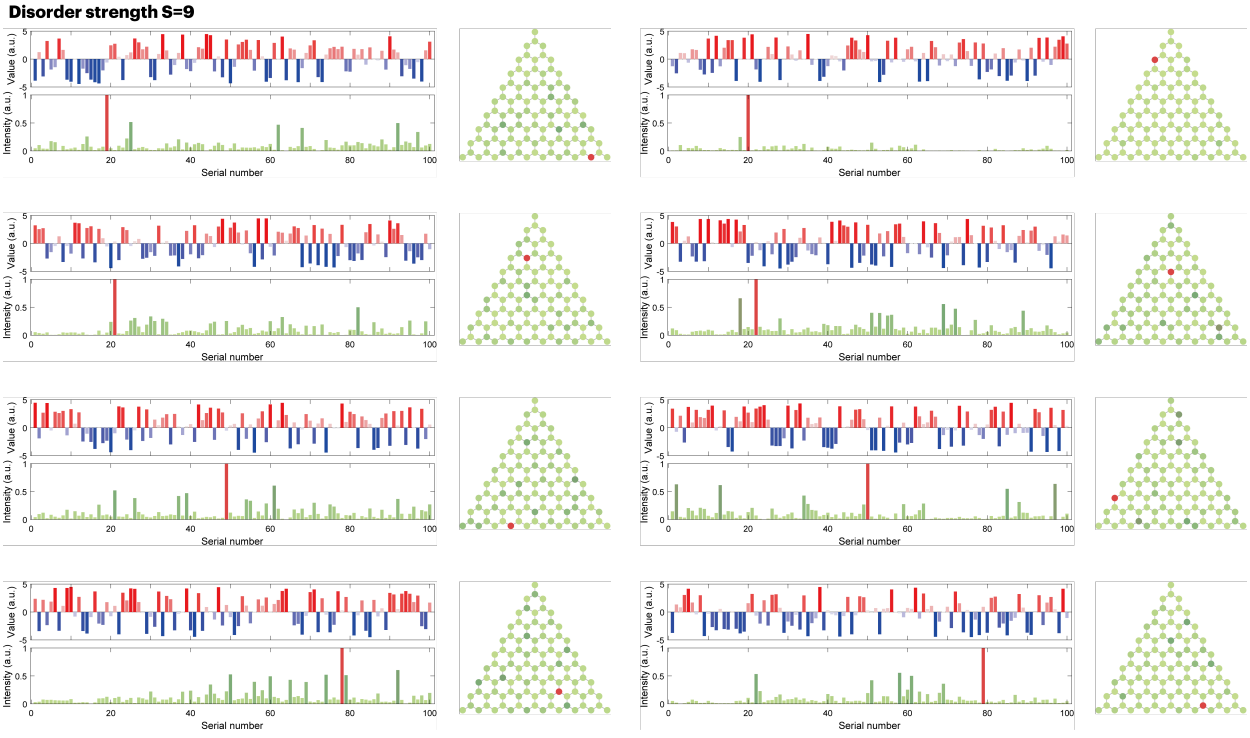

FIG. **S52. Experiment localization points with disorder strength S = 9.**

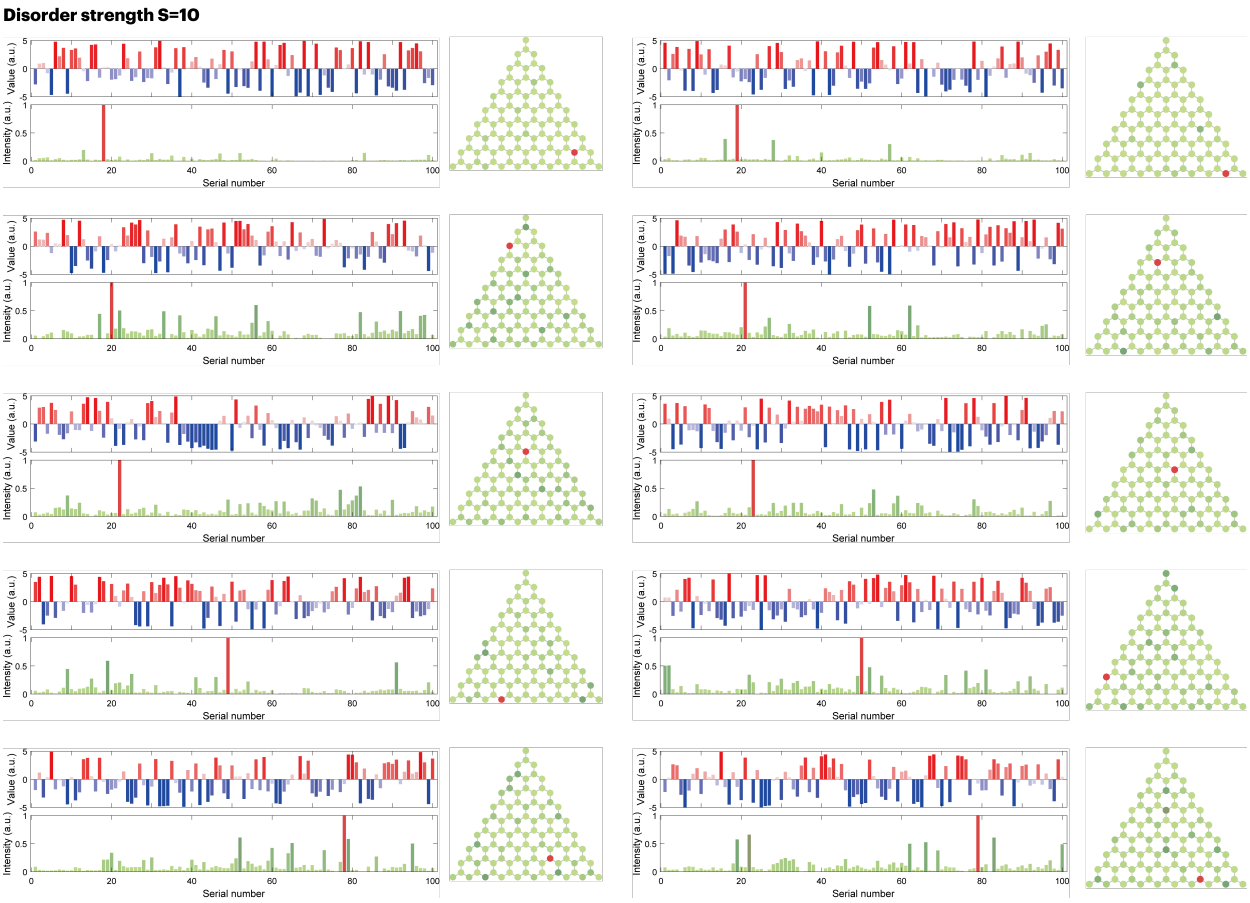

FIG. **S53**. **Experiment localization points with disorder strength S = 10.**

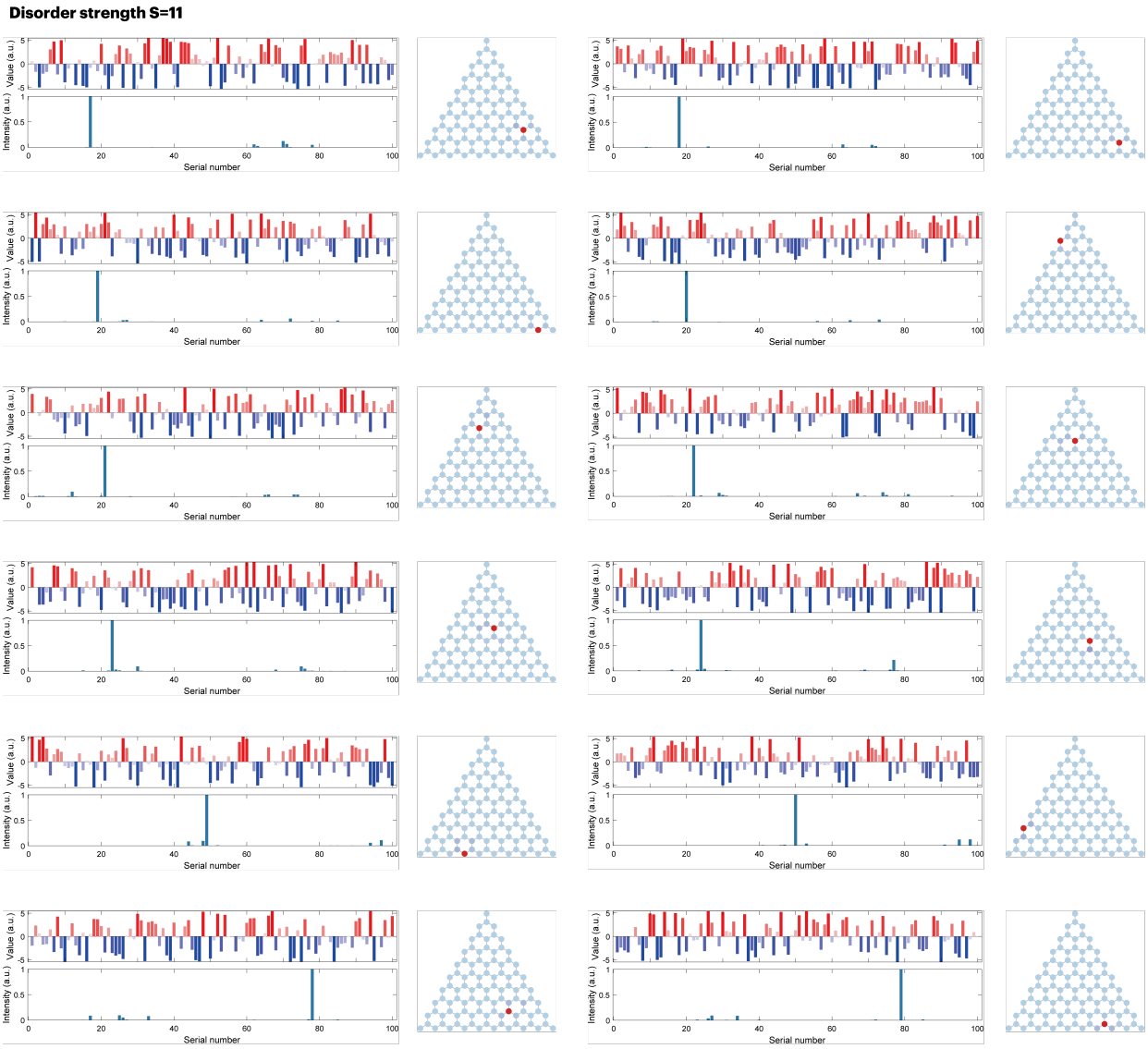

FIG. **S54. Simulation localization points with disorder strength S = 11.**

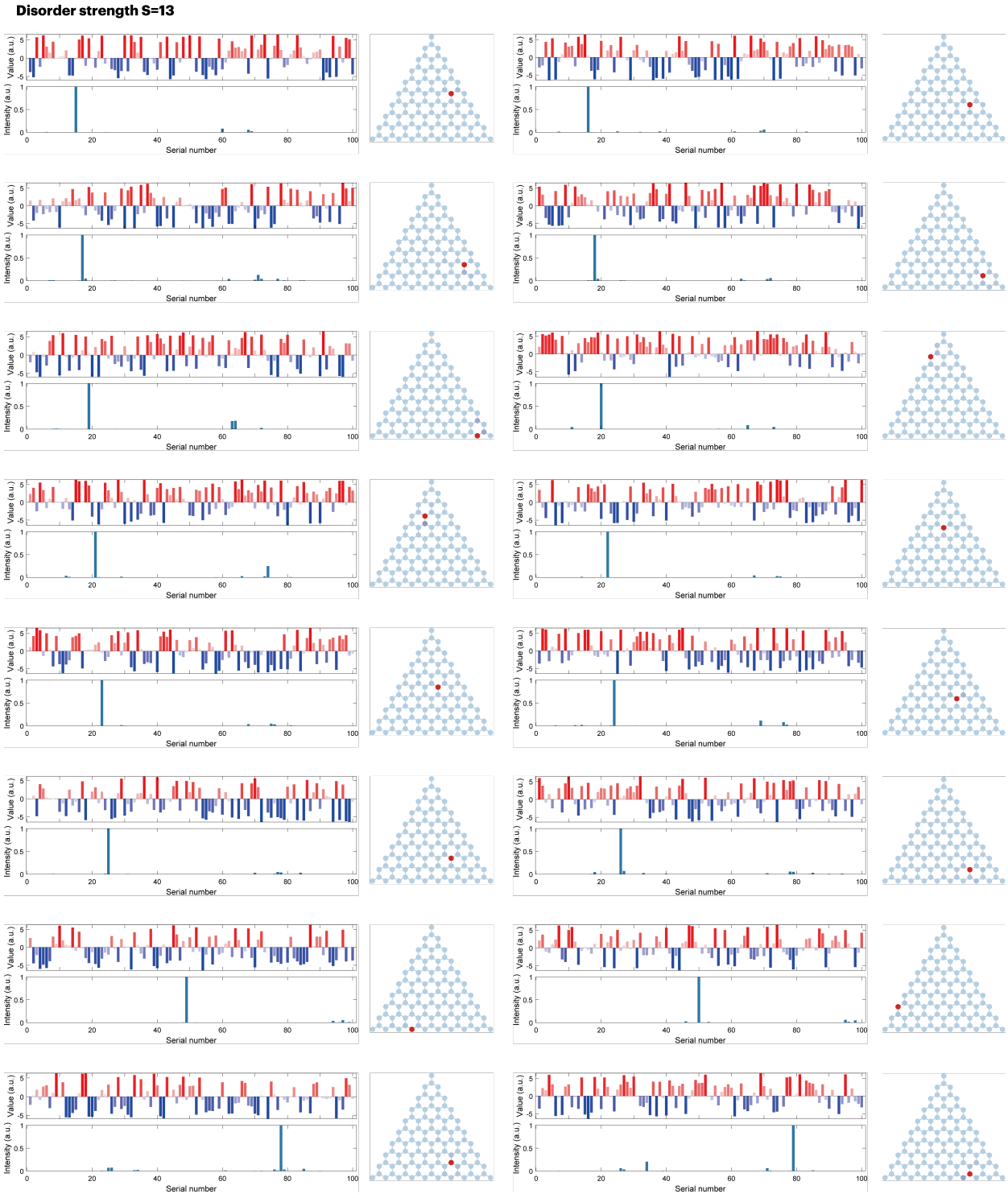

FIG. **S55. Simulation localization points with disorder strength S = 13.**

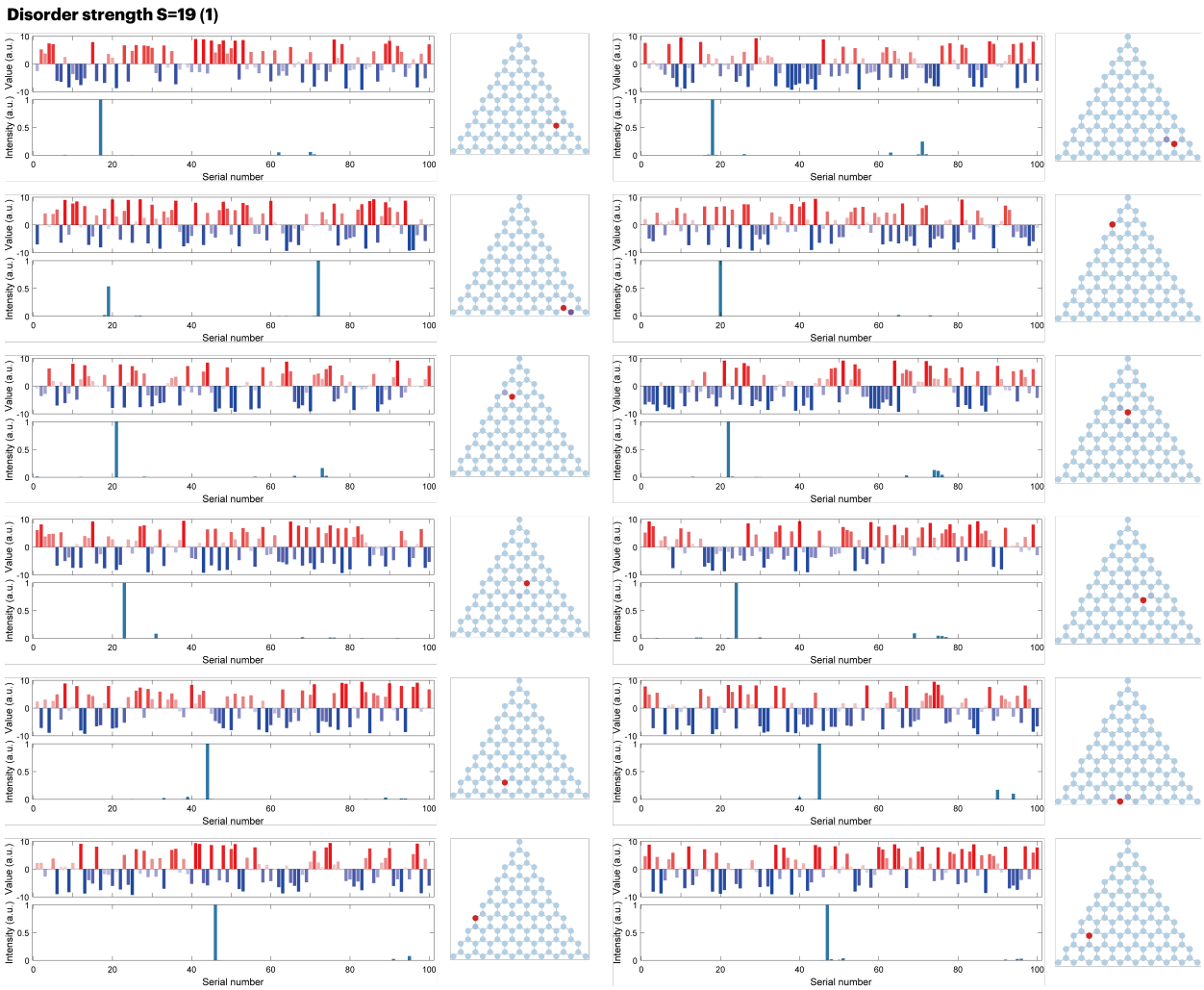

FIG. **S56**. **Simulation localization points with disorder strength S = 19 part 1.**

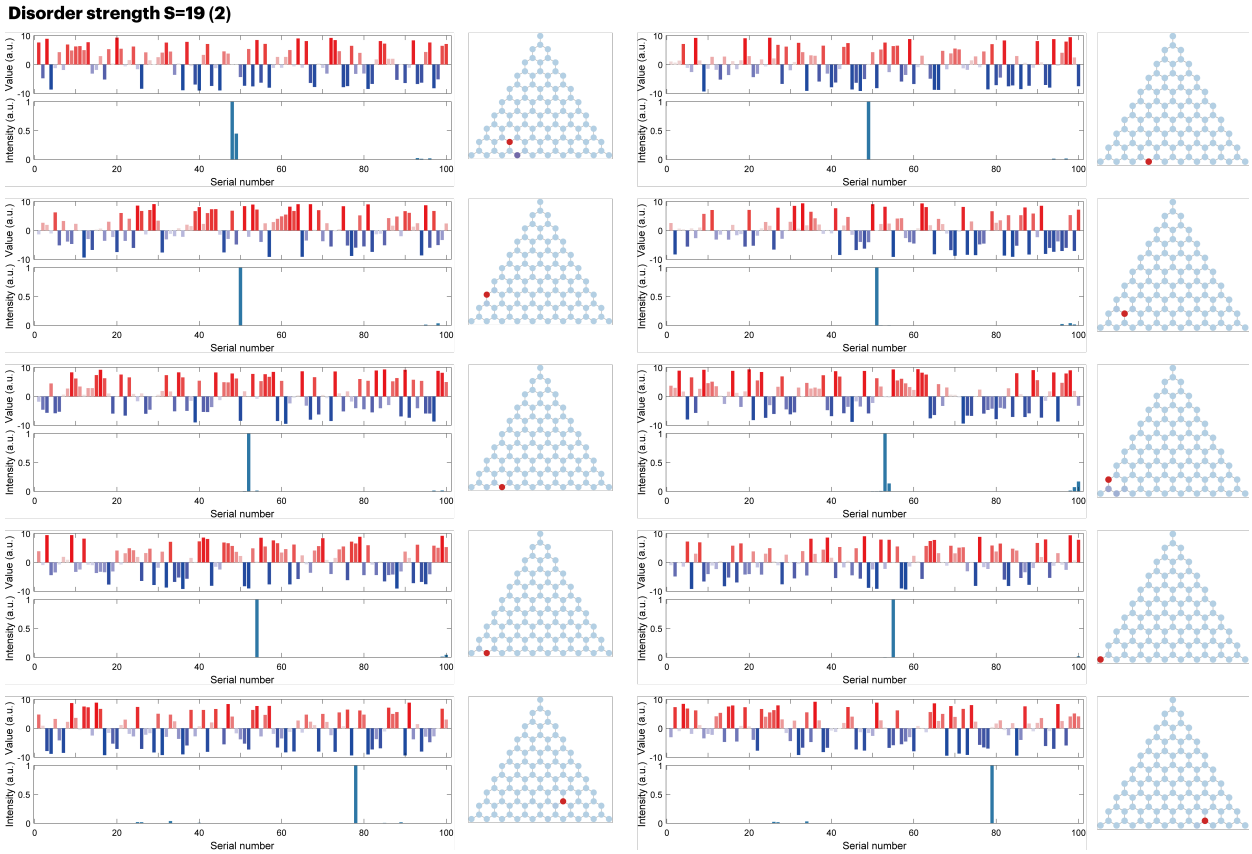

FIG. **S57. Simulation localization points with disorder strength S = 19 part 2.**

C. Discussion on the extension of Anderson hash-based probabilistic encryption framework

Because the Hash map generation process is based on Anderson Localization, which is a random disorder-induced physics effect, its feature is hardly captured or extracted by contemporary software-based algorithms such as deep neural networks. By this means, the physics-informed encryption can significantly increase information safety. In addition to the encryption framework presented in the article, we also provide several advanced encryption frameworks that leverage higher-dimensional encoding to augment information capacity.

The topology graph-based consensus Hash map can be modified from static to dynamic by implementing the modulation $G = [\mathrm{rand(V)}, \mathrm{rand(E)}]$. In this way, the random disorder rand(E) of the physics system can be dynamically changed, so that the topology graph is changed as the hopping strengths are changed. In this scenario, only when one random disorder rand(V) that encodes a character is paired with the unique graph structure rand(E), the conditions are fulfilled and the information can be decrypted. By this means, the rand(E) functions as dynamic physics uncloneable functions (PUFs). The encryption process is shown in Fig. S58.

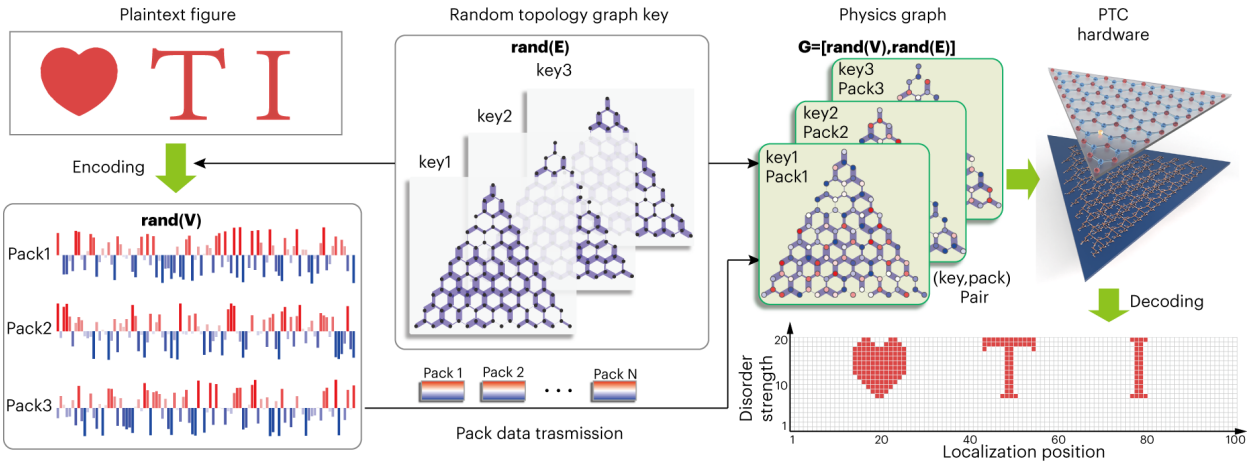

FIG. **S58**. **Extended encryption framework with dynamic tuning rand(E).** In this framework, the rand(E) decides the topological graph, thus would master the final localization state and position together with rand(V). Only when they are paired, the information can be decrypted.

D. Extension application scenario using Anderson-based probabilistic encryption framework

To further elucidate the application of the generative capability, we hereby propose another application scenario example. Anti-counterfeiting and traceability of products are long-standing priorities in commercial activities. Across disciplines, researchers have explored methods such as fluorescence-based material tagging, blockchain-based anti-counterfeiting, and even using physical unclonable functions (PUFs) as product anti-counterfeiting codes. However, even PUF-based approaches typically require recording all the anti-counterfeiting codes on the product into a database first. Later, authenticity is determined by querying the database for a match or checking the records' existence. The more products generated, the database is larger, which has a higher probability of failure. Leveraging the properties of an Anderson-localized system, we propose a new anti-counterfeiting mechanism, as illustrated in the schematic Fig. **S59**. On the manufacturer side, they encode information to be embedded (such as product type and sales/distribution channels) and map it onto node-position labels of the physical system. Using nodes as identifiers, together with a product-specific, randomly generated number sequence as a seed, the generative model produces an encrypted vector, which is printed on the product as an anti-counterfeiting code. Because the entropy space of such anti-counterfeiting code is extremely large (about 2^{128}), there can hardly be identical code generated whatever large number of products used. The products are transported and finally reach end users through various distribution channels. When authentication or traceability is required at the end user side, the user submits the product's anti-counterfeiting code via a messaging interface to an authentication agency, which loads the information onto the manufacturer-provided hardware system. From the physical response (localization characterization and spatial position), the system determines authenticity and returns it to the consumer along with ancillary information such as the distribution channel.

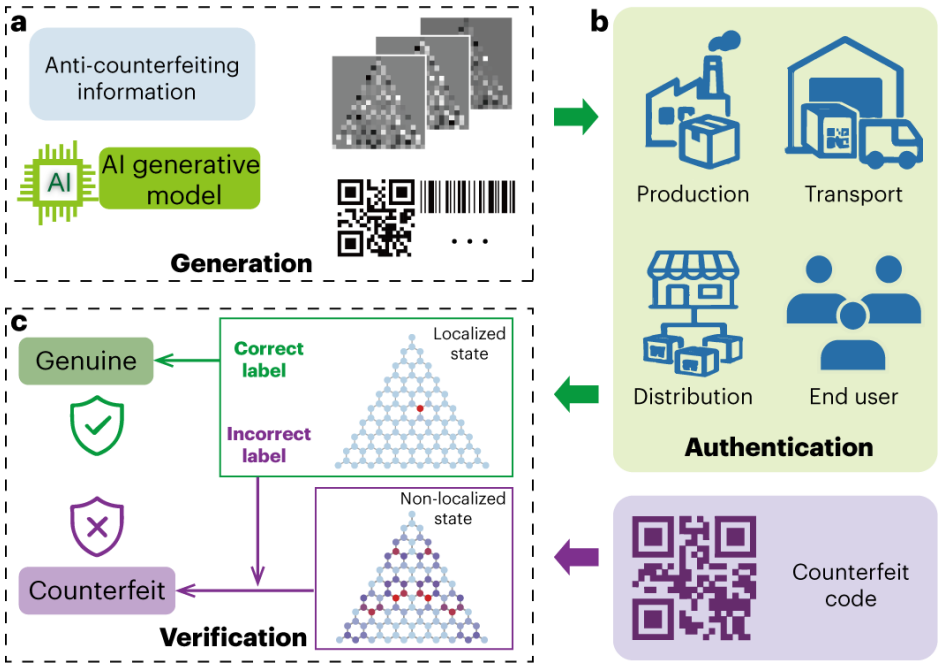

FIG. **S59**. **An example scenario of a product anti-counterfeiting application that leverages Anderson-physics characteristics and an AI-enhanced generative model.** (a) Anti-counterfeiting code preparation. The manufacturer encodes information to be embedded (such as product type and sales/distribution channels) and maps it onto node-position labels of the physical system. Using nodes as identifier inputs, together with a product-specific random seed, the generative model produces an encrypted vector, which is printed on the product as an anti-counterfeiting code. (b) Supply-chain product distribution. The products are transported and finally reach end users through various distribution channels. (c) Authentication Process. The user submits the product's anti-counterfeiting code via a messaging interface to an authentication agency, which loads the information onto the manufacturer-provided hardware system. From the physical response (localization characterization and spatial position), the system determines authenticity and returns it to the consumer along with ancillary information such as the distribution channel.

S6. DLPTC-BASED MESSAGE ENCRYPTION SYSTEM AND SECURITY PERFORMANCE ANALYSIS

Based on DLPTC system and a hash-based probability encryption framework, we proposed a practical encryption system for ASCII messages. The complete graphical user interface (GUI) of the whole system is shown in Fig. **S60**. Using the programmable to master encryption and decryption, we successfully share the message 'HelloWorld' and ASCII art 'Penrose triangle'.

In the following, we provide a security analysis framework that models the proposed cryptographic mechanism and conducts the security performance analysis.

The encryption process in this system employs a series of disordered random number vectors as the ciphertext (in the demo, a 1×100 random array). At the transmitter side, the ciphertext vector is combined with a consensus graph (i.e., the topological structure of connections among nodes) to map the plaintext to the labels of individual nodes, thereby generating the ciphertext. At the receiver side, the consensus graph data is used to extract the physical features from the random vector ciphertext, which in turn allows the recovery of the plaintext information. Thus, the consensus graph described in the system functions as the secret key (in the demo, a 1×135 vector). According to the analysis of the Anderson localization phenomenon, both the ciphertext and the key are continuous variables whose ranges are limited by the practical implementation of the system. The overall security is jointly determined by the ciphertext and the key.

A. Security and Stealthiness of Ciphertext and Key

The ciphertext and key are jointly generated by the physical topology graph system. Their generation processes are identical, as they are derived from random number sampling and share the same distribution properties. Here, they are analyzed together. For clarity, we focus on presenting the characteristics of the ciphertext. Security ensures that ciphertext and keys do not leak any meaningful information, making it impossible for attackers to infer the original data through statistical analysis, machine learning, or other methods. Stealthiness guarantees that ciphertext and keys are statistically indistinguishable from random numbers, preventing any identifiable patterns or features that could be classified or recognized. As a result, even if an adversary obtains a large set of ciphertext samples, they cannot differentiate

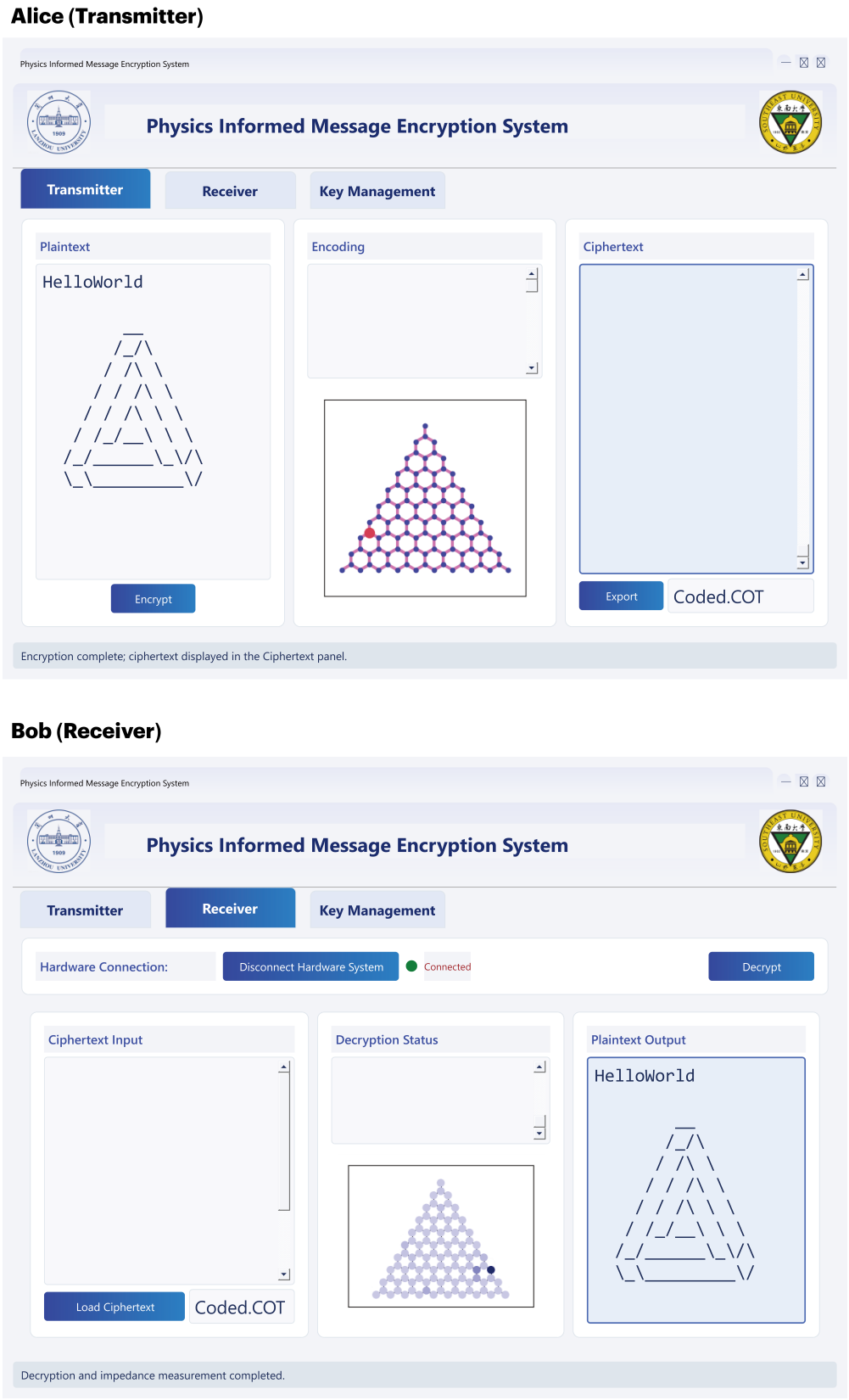

FIG. **S60. Graphical user interface (GUI) of the proposed information encryption system.**

between encoded information and random noise through statistical analysis.

- **Security-Randomness**

The system encodes information by harnessing the physical randomness generated by Anderson localization. The results are a series of random disorder vectors, whose randomness ensures that no useful statistical clues are provided to an attacker, thus satisfying the requirements for semantic security. To verify this property, we performed comprehensive statistical analyses on the generated ciphertext data (using approximately 600,000 samples) and compared it with a random array that carries no encoded information. The specific analyses include the following:

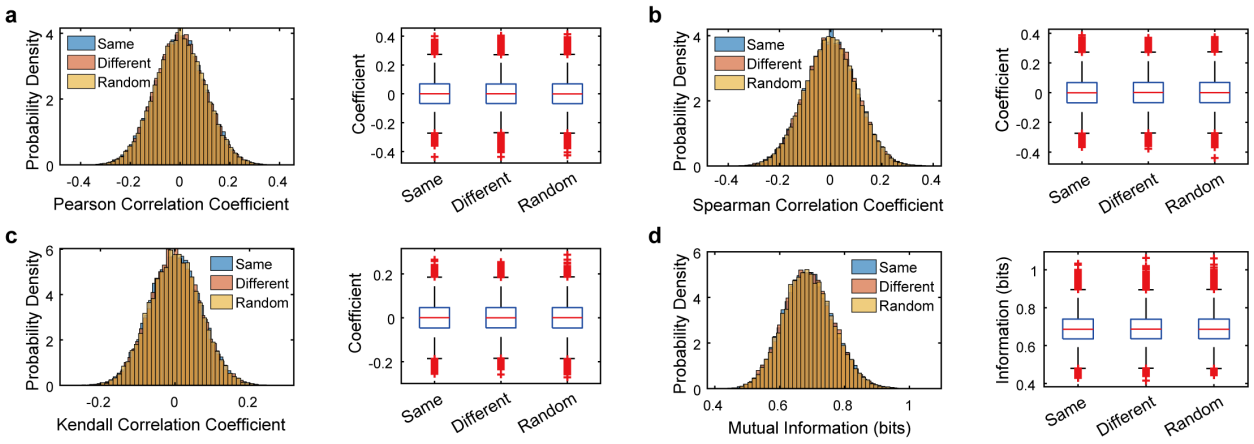

FIG. **S61**. **The histogram and boxplot of internal correlation detection.** (a) Pearson correlation coefficient. (b) Spearman correlation coefficient. (c) Kendall correlation coefficient. (d) Mutual information coefficient.

We extracted a large number of samples corresponding to **(i) the same plaintext (i.e., the same node label), (ii) different plaintexts, and (iii) random plaintexts**. For each group of data, we computed the Pearson correlation coefficient, Spearman correlation coefficient, Kendall correlation coefficient, and Mutual information coefficient. These four metrics respectively quantify the linear dependency, monotonic relationship, rank correlation, and spatial structural characteristics of the data. As can be seen from the histogram and boxplot of the Pearson correlation coefficients in Fig. **S61**a, the probability density features, main distribution positions, and ranges for the same plaintext, different plaintexts, and random numerical samples almost coincide. Similar conclusions were obtained from the analyses based on the Spearman, Kendall, and cross-correlation coefficients (see Figs. **S61**b

S61d).

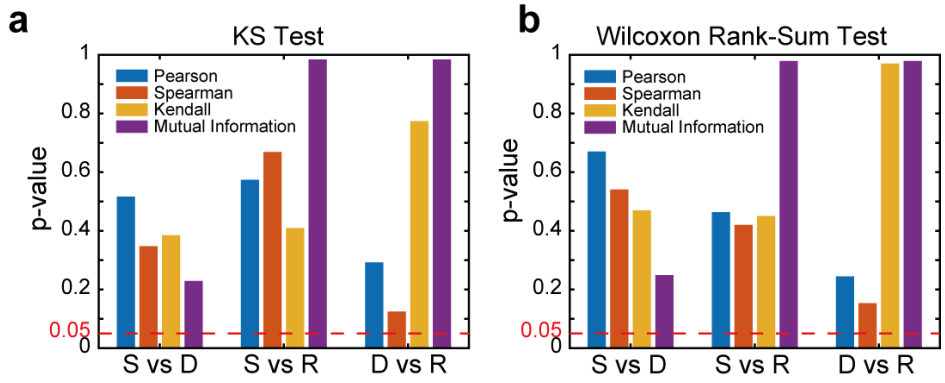

FIG. **S62**. **P-tests on statistically significant correlations.** (a) KS-test. (b) Wilcoxon Rank-sum test. S means a sample with the same label, D means a sample with a different label, and R means a pure random sample without information carried.

To further quantify these results, we performed the KS-test and Wilcoxon Rank-sum test on all four indicators (results shown in Fig. S62). All p-values far exceed the preset significance level (e.g., 0.05), indicating that no statistically significant correlations exist among the data.

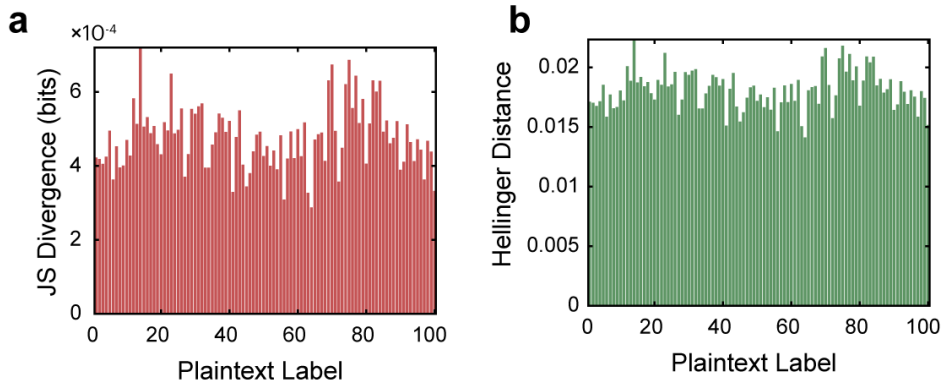

FIG. **S63**. **Uniform distribution test between data with labels and random data.** (a) JS divergence test. (b) Hellinger distance test.

- **Stealthiness-featureless and uniform in distribution**

We can use the Jensen–Shannon divergence (J-S divergence) and Hellinger distance to quantify the difference between a sample distribution and an ideal uniform distribution. The J-S divergence is a symmetric, smoothed measure of similarity between two probability

distributions, constructed based on the Kullback–Leibler divergence (K-L divergence). Due to factors such as finite samples and discretization estimation errors, when the computed value is very close (for example, between 0.01 and 0.1), we usually consider the sample distribution to be nearly identical to a uniform distribution. The Hellinger distance is a measurement of the distance between two probability distributions. Generally speaking, if the Hellinger distance is less than 0.1 or 0.2, the sample distribution is considered to be very close to uniform. After computation (results shown in Fig. **S63**), our generated information-containing samples and the uniform random numbers (without information) exhibit J-S divergences on the order of 10^{-4} and Hellinger distances on the order of 0.02 across all labels, which confirms that the statistical distribution of the ciphertext space almost perfectly coincides with that of an ideal random model.

These results indicate that even if the plaintext distribution is relatively discrete, once it is encrypted into random disorder vectors, the resulting ciphertext exhibits no discernible statistical difference from pure random noise.

B. Resistance to Brute-Force and Heuristic Clustering Attacks

When the entropy space of the system is sufficiently large, an adversary—even with certain prior knowledge—cannot effectively reconstruct or classify the data in a targeted manner. The search space remains resistant to traditional statistical methods or machine learning techniques. If classification or clustering accuracy is only equivalent to random guessing, it indicates that the system has not leaked any additional information.

- **Entropy Quantification Analysis**

We discuss the possible values of the random-number ciphertext and keys. In theory, the number of bits in the random numbers constituting the ciphertext and key can be infinitely increased; however, due to the fact that the physical state relies on actual hardware systems for data read-out, the available effective dimensionality is limited. For the Anderson system described in the paper, taking the baseline value of an edge as 1 (20pF), the allowable fluctuating range for the key is between [-5, 5]. With a maximum voltage precision of 16 bits for continuous adjustment, the average quantization precision is approximately $10/2^{16}$. Considering the response precision of the varactor diode, its variable precision is about 0.005, which we conservatively estimate with an upper bound of 0.01. If each unit is assumed to

vary independently, the maximum ciphertext entropy that the hardware supported would be $100 \times \log_2(1000) \approx 996$ bits. For the key (i.e., the variation of the edges), since its large-scale disorder can be equivalently regarded as a proportional scaling of the ciphertext's variation range, we only need to consider the smallest non-repeating increment, that is, within the range [0.5, 2]. Considering the varactor's response precision, which is about 0.01, if each unit is assumed to vary independently, the maximum key entropy that the hardware supported would be $135 \times \log_2(150) \approx 976$ bits.

When applied to our topological graph, the physical constraints caused a variation of 0.01 in precision, or a single-point variation does not significantly affect the localized state; therefore, it can be considered that when the ciphertext changes by an amount of 0.01, the plaintext remains unchanged — that is, they encode the same information. Consequently, we further analyzed the effective entropy of the system. Using the Monte Carlo method, we sampled 600,000 instances and computed the lower bound of the disorder change required to trigger a localization shift when 50% of the node values have altered. For the ciphertext, the average threshold was 0.096, and for the key, the average threshold was 0.083. To estimate the lower bound, we uniformly approximate it to be on the order of 0.1.

Under such constraints, the effective ciphertext entropy quantification problem is equivalent to the following: In a 100-dimensional space $[-5, 5]$, for any two vectors V_0 and V_1, if more than 50% of their node values differ by more than 0.1, then the two vectors are considered independent. The total number of independent vectors in the space is defined as M_{ind}. The continuous space $[-5, 5]$ is discretized at intervals of 0.1 to form the set $\mathcal{A} = \{-4.95, -4.85, \ldots, 4.85, 4.95\}$.

Each vector is composed of 100 elements (with each dimension taking only one value from $\mathcal{A}$). Thus, the total number of independent vectors is equivalent to constructing an n-dimensional, q-ary ($n = 100, q = 100$) coding problem with a Hamming distance greater than d (with $d = 51$). Moreover, to ensure that the overall vector follows a uniform distribution, the constructed code should follow Constant Composition Code (CCC) constraint [18], with the requirement that each symbol appears exactly $\lfloor n/q \rfloor = 1$ time. Suppose that a randomly chosen vector $x \in \mathcal{A}^n$ satisfies the CCC constraint (i.e., for each symbol $a_i \in \mathcal{A}$, it appears n_i times). Then, the total number of available vectors as follows:

$$|T_P| = \frac{n!}{\prod_{i=1}^{q} n_i!}\,. \tag{S32}$$

Substituting $n = 100$, we obtain $\log_2(|T_P|) \approx 524.7$ bits. Within this space, we further estimate the number of independent vectors based on the Hamming upper bound and the Gilbert–Varshamov lower bound [19, 20]. First, we construct Hamming balls with radii $r_1 = \lfloor (d-1)/2 \rfloor$ and $r_2 = d - 1$, and compute the number of vectors located within a Hamming distance at most r from a given center x. For the reference vector x, the volume of the Hamming ball is defined as:

$$\mathrm{V}(r) = \{ y \in T_P : d_H(x, y) \leq r \}, \tag{S33}$$

where $d_H(x, y)$ denotes the Hamming distance between x and any candidate independent vector y. Because directly enumerating the size of $\mathrm{V}(r)$ is computationally intractable, we employ an importance sampling method to estimate $\mathrm{V}(r)$. Fixing x, we generate perturbed vectors y from $|T_P|$ through a series of pairwise exchange operations, while ensuring that the generated codewords still belong to T_P. For a given k swaps, the total number of ways to generate a perturbation is:

$$N_{\text{way}} = \binom{n}{2k} \frac{(2k)!}{2^k k!}. \tag{S34}$$

We sample the number of swap pairs, k, from a proposal probability distribution, i.e., with the geometric distribution $p\,(1-p)^k$, the sample weight is [21]:

$$w_i = \frac{N_{\text{way}}}{p\,(1-p)^k}. \tag{S35}$$

After performing N_{sample} samples, the weighted proportion of samples satisfying $d_H(x, y) \leq r$ is:

$$\rho(r) = \frac{\sum_{i=1}^{N_{\text{sample}}} w_i \, \mathbf{1}\{d_H(x, y_i) \leq r\}}{\sum_{i=1}^{N_{\text{sample}}} w_i}, \tag{S36}$$

where $\mathbf{1}\{d_H(x, y_i) \leq r\}$ is the indicator function that equals 1 if $d_H(x, y_i) \leq r$ and 0 otherwise. Since $\rho(r)$ represents the probability that a randomly selected vector (from the CCC space) has a Hamming distance at most r, the volume of the Hamming ball is:

$$V(r) = |T_P| \times \rho(r). \tag{S37}$$

Based on this, we use the Hamming upper bound and the Gilbert–Varshamov lower bound from coding theory to estimate the order of magnitude of the problem. Using a radius of $r_1 = \lfloor (d-1)/2 \rfloor$ to estimate the Hamming upper bound gives:

$$\mathrm{UB}(M_{\text{ind}}) \approx \frac{|T_P|}{V(r_1)} = \frac{|T_P|}{|T_P| \cdot \rho(r_1)} = \frac{1}{\rho(r_1)}. \tag{S38}$$

Similarly, using a radius of $r_2 = d-1$ to estimate the Gilbert–Varshamov lower bound gives:

$$\mathrm{LB}(M_{\mathrm{ind}}) \approx \frac{|T_P|}{V(r_2)} = \frac{|T_P|}{|T_P| \cdot \rho(r_2)} = \frac{1}{\rho(r_2)}\,. \tag{S39}$$

Using the importance sampling method with 10^9 sampling numbers, for the ciphertext within $[-5, 5]$, the upper bound on the ciphertext entropy is $\log_2[\mathrm{UB}(M_{\mathrm{ind}})]$ bits, and the lower bound is $\log_2[\mathrm{LB}(M_{\mathrm{ind}})]$ bits.

For the key, there are 135 edge connections, and each connection can take values in the range $[0.5, 2]$. This is equivalent to constructing a q-ary Constant Composition Code with n=135 dimensions and q=15, and a Hamming distance greater than $\lceil 0.5 \times 135 \rceil = 68$. Using the same method, one can estimate the upper bound and lower bound of the key entropy. Here we show the value on different scales in **S64**. It can be seen that under scale $N = 9$, the entropy value of ciphertext and key can reach approximately 128 bits. And they increase faster as the scale increases. At $N = 20$, it can be over 500 bits, which is far more than the current software-based algorithm.

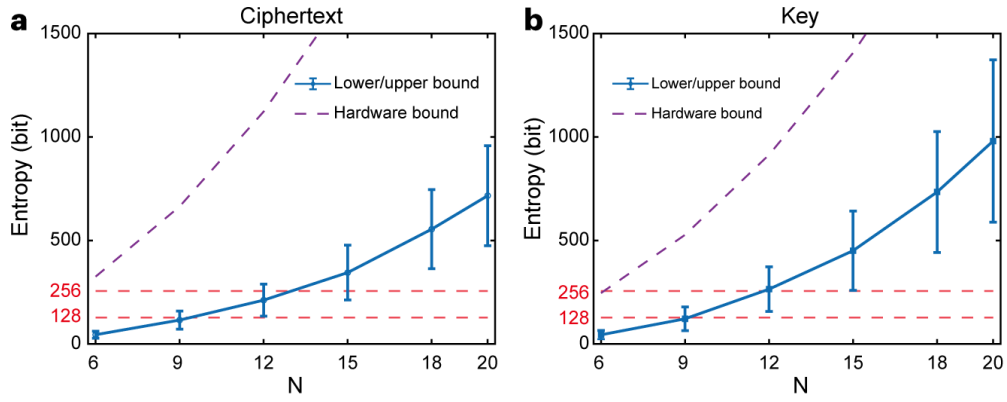

FIG. **S64**. **The hardware supported entropy and the physics system supported entropy estimation** (a) The ciphertext entropy. (b) The key entropy.

It is important to note that this estimate is rather conservative:

1. The discrete set $\mathcal{A} = \{-4.95, -4.85, \ldots, 4.85, 4.95\}$ is clearly much smaller than the actual continuous space available in $[-5, 5]$.

2. Even if the Hamming distance between two vectors is less than 51, there is still a nonzero probability that the plaintext will change (for example, two vectors with a Hamming distance of 20 may still represent independent ciphertexts/keys); this possibility is not taken into account in the above computation.

3. The Constant Composition Code requirement here mandates that each symbol appears exactly once, whereas a uniformly random distribution may not satisfy the CCC constraint, which may lead to an underestimation. Thus, our estimate is very conservative.

- **Attack Models and Performance Evaluation**

The core idea behind the ciphertext's resistance to attacks is that even if an attacker knows that the random disorder vectors map to a limited set of 100 node labels, they would be forced to resort to random guessing (with a success probability of 1% per node) in the absence of additional information. The entropy is large enough so that a brute-force attack is almost impossible. As a result, we give two machine learning based heuristic attacks.

Figure **S65**a illustrates the result of Uniform Manifold Approximation and Projection (UMAP) for dimensionality reduction and visualization on the generated data. UMAP approximates the local relationships in high-dimensional data as a graph structure and then maps this graph into a lower-dimensional space through an optimization process, thereby facilitating feature analysis. It can be observed that for any displayed label, the corresponding data are uniformly dispersed in the 2D space, and samples with the same label do not exhibit a clustering tendency.

Next, we assume that the plaintext consists of 100 classes. We attempt to attack the relationship between plaintext and ciphertext by first capturing the underlying distribution from the ciphertext via unsupervised learning and then recovering the plaintext labels using a matching algorithm. Initially, a Gaussian Mixture Model (GMM) is applied to the ciphertext samples to perform clustering. The model provides a "soft" clustering outcome, that is, a probability for each data point belonging to each cluster. At this stage, a cost matrix is constructed where the cost represents the error or inconsistency incurred when matching different clusters to the plaintext labels. By employing the Hungarian algorithm (also known as the Kuhn-Munkres algorithm), we solve this minimum-cost assignment problem to obtain the optimal correspondence between the GMM clustering results and the plaintext labels. With this optimal matching, mapping the GMM-assigned labels to the corresponding plaintext labels allows us to preliminarily restore the correspondence between plaintext and ciphertext in the encryption system. The outcome of this attack is shown in Figs **S65**b and **S65**c. The 100 clusters each contain a nearly uniform number of samples, and the predicted

accuracy is only 1.65%, which is close to the random guessing accuracy of 1%. This indicates that no feature correlation between the data and the plaintext has been extracted.

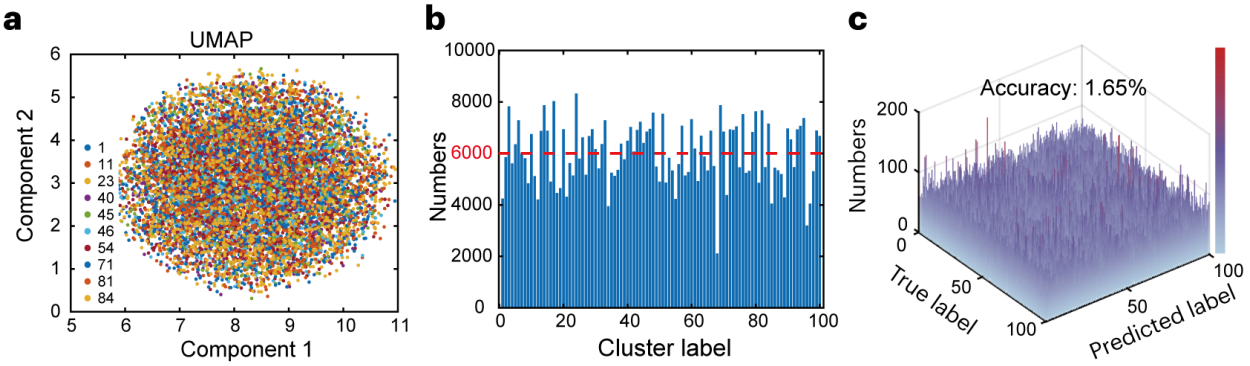

FIG. **S65**. **Dimensionality reduction and heuristic clustering attack.** (a) UMAP dimensionality reduction and visualization for samples. (b) GMM clustering results on all labels. (c) Confusion matrix of the predicted labels.

Subsequently, we further enhanced the attack process by training an unsupervised autoencoder that reduces the input samples to a 50-dimensional latent space while automatically extracting features. After the network converges, clustering analysis is performed on the latent space using the same GMM combined with the Hungarian algorithm. The results, presented in Fig. **S66**, yield a classification accuracy of 1.66%, which is consistent with the previous result of 1.65% obtained directly from feature extraction. This further demonstrates that no additional information could be derived, and the feature correlation between the data and the plaintext remains unattainable.

These results indicate that the system exhibits robust resistance to brute-force and heuristic clustering attacks.

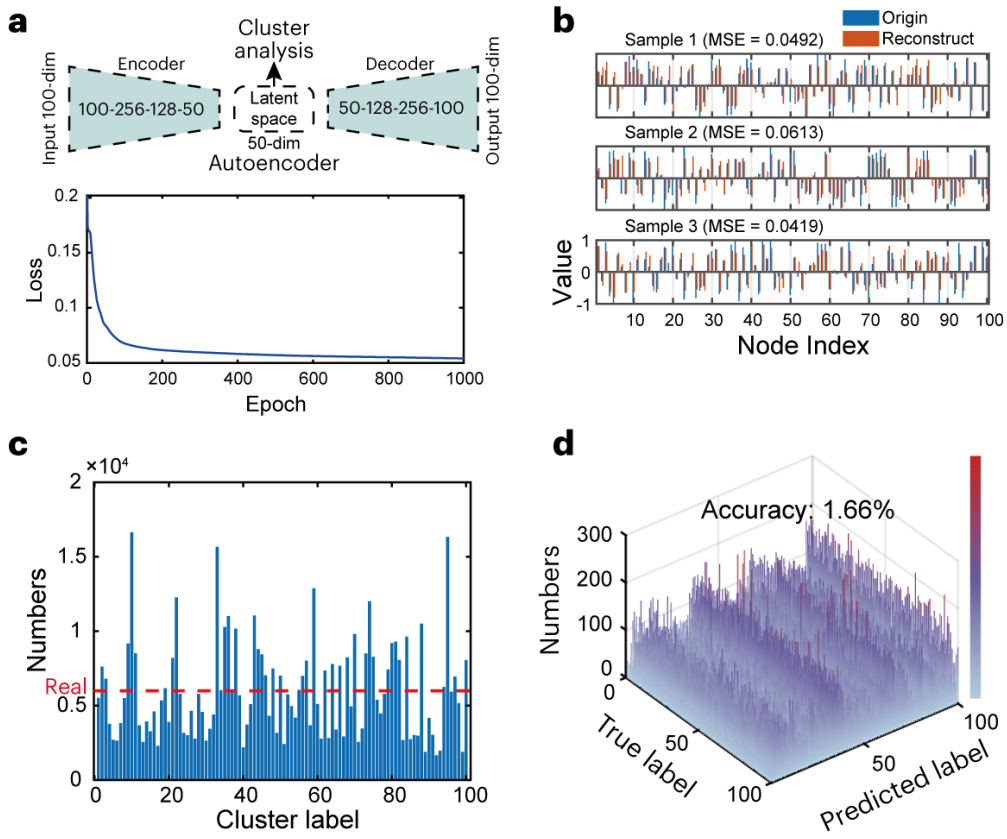

FIG. **S66**. **Dimensionality reduction and heuristic clustering attack.** (a) The pipeline and training process of the autoencoder. (b) The reconstruction results produced by the trained autoencoder. (c) GMM clustering results on all labels. (d) Confusion matrix of the predicted labels.

In summary, considering the dimensions of randomness, entropy space, and resistance to brute-force and heuristic clustering attacks, our system exhibits excellent security.

C. Flexible and Scalable Security Enhancement Mechanism

- **Dynamic Consensus Mechanism**

The system is capable of employing a dynamic consensus graph as the key. This is based on the fully programmable nature of our DLPTC system and the impact shown in Fig. 4 of the main manuscript, where variations in rand(E) (i.e., changes in the consensus graph) influence the state and position of Anderson localization. By regulating rand(E) to achieve dynamic key management, each individual or group of ciphertexts is protected by an

independent dynamic key. Consequently, even if an attacker intercepts a single ciphertext, breaking it remains extremely difficult because the attacker faces a one-time key protecting only a brief message. This design offers clear forward secrecy, providing strong theoretical support for key security.

- **Expansion of the Graph Scale**

In the current demo system (N=9), the consensus graph comprises 100 nodes. We further studied the impact of enlarging the lattice scale on system security. For instance, when N=15, the number of nodes increases to 256; when N=20, the number reaches 441. The consensus graph scale increases, the overall system security is further enhanced, and the number of possible mappings that an attacker would have to search exhaustively also increases.

- **Ciphertext Diffusion Coding**

Currently, the mapping space for the plaintext is limited to 100 node labels, restricting the plaintext's entropy. To address this, plaintext group diffusion coding can be applied such that a slight change in the plaintext affects the entire group's encoding result. In this scenario, the whole group is regarded as a single message, and its entropy reflects the product of the coding possibilities within the group. This method effectively increases the entropy of both the plaintext and the ciphertext, providing additional security assurance for the system.

- **Enhanced Circuit System Design**

The adjustable range of the present circuit system limits the analysis of the entropy space; however, on an advanced circuit system platform with a larger adjustable range, the entropy spaces of both the ciphertext and the key can be further expanded, thereby enhancing the overall security of the system.

- **Entropy Loss and Semantic Security**

Although in theory, some loss of entropy occurs when mapping from a random ciphertext with infinite entropy to a plaintext with a finite state space, such a nonlinear mapping is an inherent aspect of encryption function design. As long as the mapping is noninvertible to an attacker who does not possess the key and the ciphertext is statistically indistinguishable from pure random noise (thus satisfying semantic security requirements), the entropy loss does not present a security risk.

The main conclusions are summarized as follows:

1. **Security and Stealthiness of Ciphertext and Key:** Through statistical analysis of the Pearson, Spearman, and Kendall correlation coefficients, as well as the mutual information coefficient, and accompanying K-tests, we demonstrate that the probability–density features, primary distribution loci, and value ranges for identical plaintexts, different plaintexts, and random data virtually coincide. No statistically significant correlations are detected. Furthermore, tests based on the Jensen–Shannon divergence and the Hellinger distance confirm that these distributions are statistically indistinguishable from those of pure random numbers.

2. **Resistance to Brute-Force and Heuristic Clustering Attacks:** By means of quantitative entropy analysis, the system's effective entropy for both ciphertext and key already exceeds 128 bits at scale $N = 9$. As the system scale increases, entropy grows rapidly, reaching approximately 1000 bits at $N = 20$—thereby ensuring robust security margins. In addition, dimensionality-reduction and self-supervised clustering attacks (e.g., autoencoder-based attacks) show that even with known-class priors, classification accuracy remains no better than random guessing, so that it can effectively resist the attacks.

3. **Strong Scalability:** We discuss several mechanisms—Dynamic Consensus, Graph-Scale Expansion, and Ciphertext Diffusion Coding—that can be employed to further elevate the system's performance and security as the network grows.

In summary, the encryption system constructed based on physical randomness and a dynamic consensus mechanism demonstrates high security in terms of ciphertext randomness, resistance to brute-force and heuristic clustering attacks, and key security. As a laboratory demo, the system performs well in experimental settings; however, for commercial cryptographic applications, further enhancements in key distribution, side-channel protection, and other aspects are necessary.

---

[1] C. H. Lee, S. Imhof, C. Berger, F. Bayer, J. Brehm, L. W. Molenkamp, T. Kiessling, and R. Thomale, "Topolectrical circuits," Commun. Phys. **1**, 39 (2018).

[2] J. He, H. Jia, H. Chen, T. Wang, S. Liu, J. Cao, Z. Gao, C. Shang, and T. J. Cui, "Mode engineering in reconfigurable fractal topological circuits," Phys. Rev. B **109**, 235406 (2024).

[3] T. Li, P. Zhu, W. A. Benalcazar, and T. L. Hughes, "Fractional disclination charge in two-dimensional C_n-symmetric topological crystalline insulators," Phys. Rev. B **101**, 115115 (2020).

[4] C. W. Peterson, T. Li, W. A. Benalcazar, T. L. Hughes, and G. Bahl, "A fractional corner anomaly reveals higher-order topology," Science **368**, 1114 (2020).

[5] C. W. Peterson, T. Li, W. Jiang, T. L. Hughes, and G. Bahl, "Trapped fractional charges at bulk defects in topological insulators," Nature **589**, 376 (2021).

[6] J. Yang, Y. Li, Y. Yang, X. Xie, Z. Zhang, J. Yuan, H. Cai, D.-W. Wang, and F. Gao, "Realization of all-band-flat photonic lattices," Nat. Commun. **15**, 1484 (2024).

[7] B. Xie, W. Deng, J. Lu, H. Liu, P. Lai, H. Cheng, Z. Liu, and S. Chen, "Correspondence between real-space topology and spectral flows at disclinations," Phys. Rev. B **108**, 134118 (2023).

[8] N. Murphy, R. Wortis, and W. Atkinson, "Generalized inverse participation ratio as a possible measure of localization for interacting systems," Phys. Rev. B **83**, 184206 (2011).

[9] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," J. Big Data **6**, 1 (2019).

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**, 436 (2015).

[11] D. P. Kingma, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 (2014).

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," J. Mach. Learn. Res. **15**, 1929 (2014).

[13] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," Adv. Neural Inf. Process. Syst. **33**, 6840 (2020).

[14] T. Salimans and J. Ho, "Progressive distillation for fast sampling of diffusion models," arXiv preprint arXiv:2202.00512 (2022).

[15] J. Ho and T. Salimans, "Classifier-free diffusion guidance," arXiv preprint arXiv:2207.12598 (2022).

[16] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," Adv. Neural Inf. Process. Syst. **2**, 3483 (2015).

[17] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, Vol. 30 (Atlanta, GA, 2013) p. 3.

[18] I. Csiszár and J. Körner, *Information theory: coding theorems for discrete memoryless systems* (Cambridge University Press, 2011).

[19] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*, Vol. 16 (Elsevier, 1977).

[20] R. Roth, *Introduction to coding theory* (Cambridge University Press, 2006).

[21] C. P. Robert, G. Casella, and G. Casella, *Monte Carlo statistical methods*, Vol. 2 (Springer, 1999).