RESEARCH ARTICLE | SEPTEMBER 10 2024

Metalens array for quantum random number

Special Collection: Quantum Metamaterials

Yubin Fan ; Shufan Chen ; Xiaoyuan Liu ; Xiaoyu Che ; Xiaodong Qiu ; Mu Ku Chen ; Din Ping Tsai

Appl. Phys. Rev. 11, 031418 (2024)
https://doi.org/10.1063/5.0224766

Applied Physics Reviews ARTICLE pubs.aip.org/aip/are

Metalens array for quantum random number

Cite as: Appl. Phys. Rev. **11**, 031418 (2024); doi: 10.1063/5.0224766
Submitted: 21 June 2024 · Accepted: 20 August 2024 ·
Published Online: 10 September 2024

View Online Export Citation CrossMark

Yubin Fan,[1,2,3] Shufan Chen,[1,2,3] Xiaoyuan Liu,[1,2,3] Xiaoyu Che,[1,2,3] Xiaodong Qiu,[1,2,3] Mu Ku Chen,[1,2,3] and Din Ping Tsai[1,2,3,a)]

AFFILIATIONS

[1]Department of Electrical Engineering, City University of Hong Kong, Kowloon, Hong Kong 999077, People's Republic of China
[2]Centre for Biosystems, Neuroscience, and Nanotechnology, City University of Hong Kong, Kowloon, Hong Kong 999077, People's Republic of China
[3]The State Key Laboratory of Terahertz and Millimeter Waves, City University of Hong Kong, Kowloon, Hong Kong 999077, People's Republic of China

Note: This paper is part of the APR Special Topic on Quantum Metamaterials.
[a)]**Author to whom correspondence should be addressed:** dptsai@cityu.edu.hk

ABSTRACT

Quantum random number generation (QRNG) leveraging intrinsic quantum uncertainty has attracted significant interest in the field of integrated photonic architecture, with applications in quantum cryptography, tests of quantum nonlocality, and beyond. The demand for compact, low-energy consumption, robust, fast, and cost-effective QRNGs integrated into photonic chips is highlighted, whereas most previous works focused on bulk optics. Here, based on the metalens array entangled source, we experimentally realized a miniaturized, high-dimensional quantum random number generator via a meta-device without post-randomness extraction. Specifically, the device has a high-density output with 100 channels per square millimeter. This chip-scale quantum randomness source can obtain random number arrays without post-randomness extraction and enable compact integration for quantum applications needing secure keys or randomness. Our approach demonstrates potential in secure key generation and randomness for quantum applications.

INTRODUCTION

In the digital era, the safeguarding of sensitive information represents a critical challenge across industries and sectors, such as finance, healthcare, government, and technology, where the integrity and confidentiality of data are paramount. As traditional encryption methods increasingly succumb to advanced and artificial intelligence (AI)-driven hacking techniques,[1] the urgency for implementing enhanced security measures has intensified. Quantum random numbers, derived from the principles of quantum mechanics, offer a robust solution to these vulnerabilities. These numbers, generated through quantum random number generators, are inherently unpredictable and truly random, making them ideal for fortifying encryption and authentication systems against cyberattacks.[2,3] With the rise of big data, cloud computing, and advanced AI, the integration of quantum random numbers into data security protocols is not only advantageous but also essential. This quantum approach not only promises to revolutionize the landscape of information security but also aligns with the increasing complexity and demands of global data protection standards.

The development of quantum technologies drives the demand for quantum random numbers in applications such as quantum key distribution (QKD)[4,5] and verification of Bell's inequality.[6] Random numbers are sequences of numbers that are unpredictable and independent. They are uniformly distributed over a given range, meaning that each number has an equal probability of occurrence. To verify the basic physical principles of quantum mechanics, Ture random number generators (TRNGs) are developed such as cosmic photons' arrival times offer an avenue for generating random numbers, not only as a source of randomness but also for closing the freedom-of-choice loophole in quantum nonlocality tests.[7] Quantum mechanical phenomena, like the uncertainty principle and entanglement, are a promising approach for generating truly random numbers. Device-independent Quantum random number generations (QRNGs) provide random numbers with the highest security among TRNGs. For example, source-device-independent quantum random number generation leverages the arrival time of photons from untrusted entangled sources and the nonlocal dispersion cancelation effect for heightened security and high-speed random number generation.[8] Device-independent

QRNGs provide high security in complex setups. QRNGs based on simpler trusted devices can be applied to more scenarios. Practical high-speed quantum random number generators have emerged, using the timing of single-photon detection as raw data and achieving impressive bit rates of 109 Mbps after bias reduction and randomness extraction.[9] Another applicable way to generate QRNG from trusted device is photon number parity verification, using multiplexed transition-edge sensors to resolve up to 100 photons and generate unbiased random numbers based on coherent state parity.[10] Random number generation is a core component of quantum technologies. In the past, most work was based on bulky optical elements and random number generators. Miniaturized high-dimensional QRNG devices have the potential to be used in personal communication terminals based on quantum technologies, such as communication protocols containing Bell's inequality verification through multiple channels in personal terminals. Quantum optical chips are indeed a crucial area with many uses,[11] including continuous-variable QKD.[12] The demand for compact, low-energy consumption, robust, fast, and cost-effective QRNGs integrated into photonic chips is highlighted.[13,14]

Metasurfaces enable flexible control over light wavefronts and have been integrated into compact quantum devices for applications such as quantum state preparation and modulation, placing demands for further miniaturized quantum components like sources of true randomness. Metasurfaces are a kind of high-performance platform composed of subwavelength antennas. They are compact and easy to integrate[15–30] and are widely used in the context of quantum research.[31–40] Very recently, the learned metasurfaces for 360° structured light,[25] have achieved a remarkable technological breakthrough in 3D imaging and holographic projection and thus, opening new horizons for applications in photonics interaction and high-dimensional quantum technology. The realization of high-dimensional quantum light sources has achieved tremendous success. Standing on the shoulders of giants,[31] we have developed a realization of a high-dimensional quantum random number generator in a miniaturized form factor. We experimentally realized a high-dimensional quantum random number generator based on our metalens array. We propose a high-dimensional random number generator leveraging a high-dimensional entangled photon source. We harnessed a quantum random number array by collecting photons emitted from a β barium borate (BBO) crystal pumped by a continuous-wave (CW) laser diode with arrival time differences following a Poisson distribution. Utilizing a metalens array combined with spontaneous parametric downconversion (SPDC) in the BBO crystal, we recorded the time differences of photon arrivals, obtaining a high-dimensional quantum random number array without post-randomness extraction. AI systems, while improving the efficiency of information processing, can also be exploited by malicious actors to pose threats to information security. Our generated sequence of random numbers can pass NIST randomness tests and resist attacks against a Generative Adversarial Network (GAN) model, as Fig. 1 shows. In this context, it is crucial to proactively address information security challenges by establishing a robust information security protection system. This involves not only advancing the adoption of quantum random numbers but also examining the potential security threats posed by AI in the information security domain. Only by taking a technology-driven approach can we build a trustworthy digital world that instills confidence in its users.

RESULTS

The working principle of high-dimensional quantum random number generator

The generation principle is that continuous laser photon arrival times obey Poisson distribution, and the specific arrival times between two photons are random.[9] SPDC is a random, nonlinear process. When the number of photons within the correlated time is much less than 1, the SPDC process also obeys Poisson distribution.[41] These two random occurrences are the basis for generating our high-dimensional random number arrays. The two-photon state from the metalens array can be written as[31]

$$\psi = \frac{1}{\sqrt{n}}(|0,0\rangle + |1,1\rangle + |2,2\rangle + \cdots + |n-1,n-1\rangle), \quad (1)$$

where n is the total number of metalens involved in the random number generation, defined as the number of dimensions.

The details of the numerical simulation and fabrication of the metalens array are shown in supplementary material Note 1. In our QRNG array scheme, the CW laser focuses through a metalens array into a BBO crystal to generate nonlinear effects and form an SPDC photon pair array. The experimental setup is shown in Fig. 2(a). See more details in the supplementary material Note 2. A CW laser passes through a metalens array to generate an SPDC photon pair array. The number of arrival photons is less than one in each reference period T_{ref}. The photon is detected by a Single photon counting module (SPCM, SPCM-800-14-FC). The SPCM has a dead time of 22 ns and a maximum count rate of 37 Mcps with continuous light illumination. To measure the timing of photon arrival, we use high-performance timing measurement electronics with a time-to-digital converter (TDC, SIMINICS FT1040) operating with a time resolution of 128 ps. The arrival time transfers the original random bits into the personal computer (PC) through USB3.0 and then uses the selected reference time as the "start" of TDC and the SPCM detection signal as "stop" to record which time bin the photon arrives at and to finally output a random number.

For an ideal uniform distribution, the probability of a photon detection falling into a certain N_{b} bin is $1/N_{\mathrm{b}}$, each time bin has a time

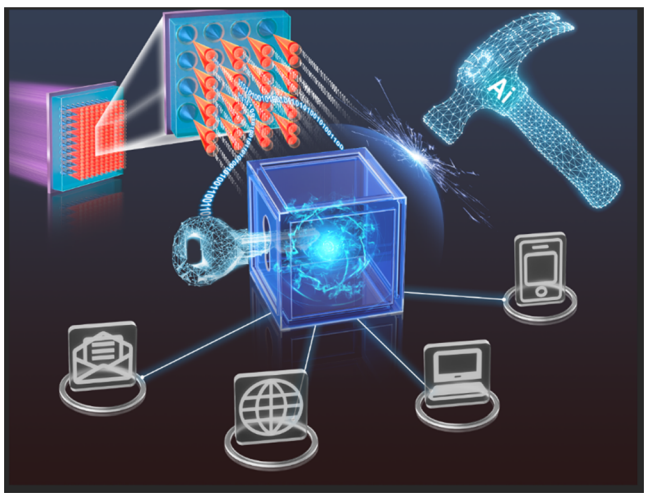

FIG. 1. Schematic of metalens array for quantum random number.

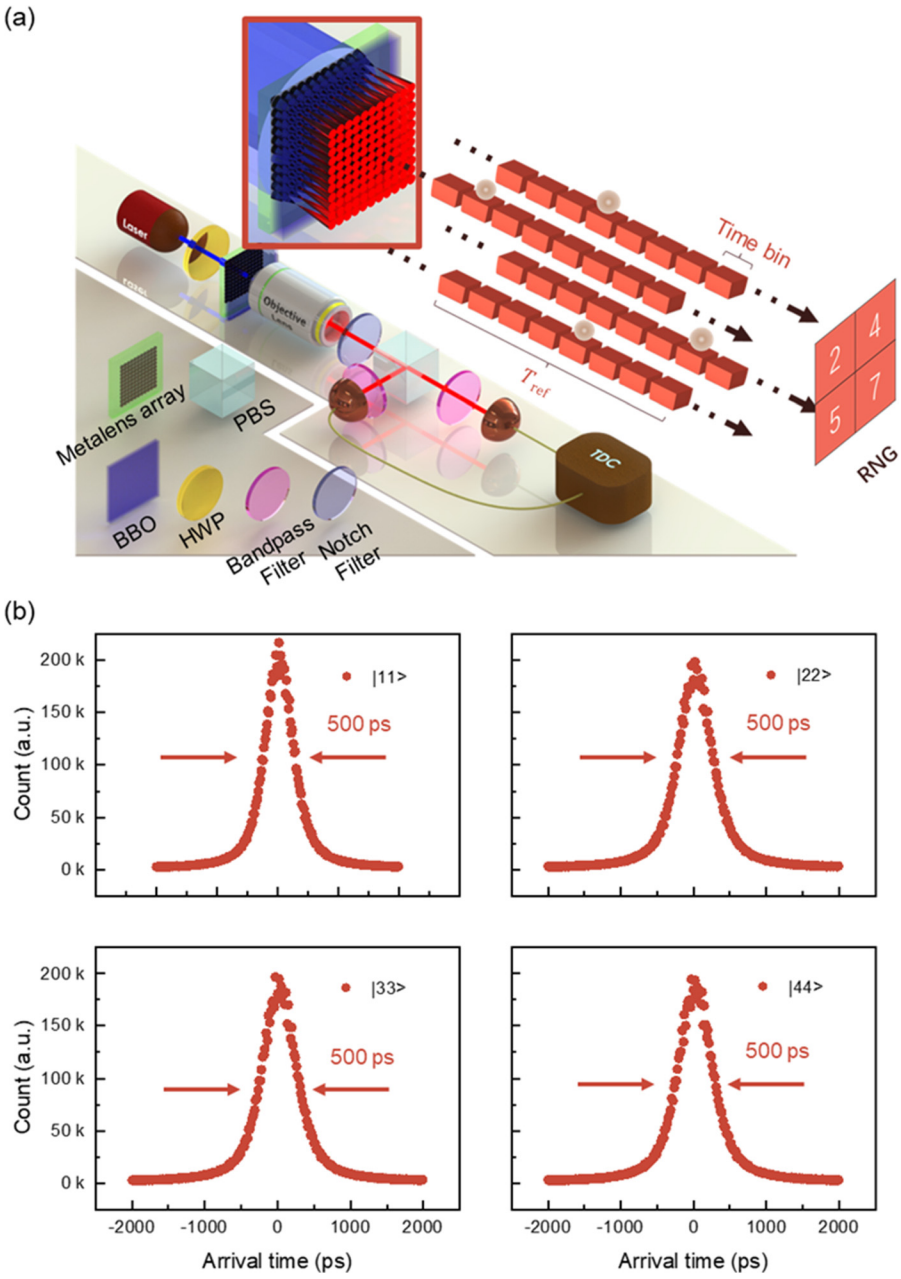

FIG. 2. Setup and direct counting results. (a) Experimental setup of high-dimensional quantum random number generator. (b) Time difference statistics of two SPDC photons emitted from a metalens.

duration T_b. The resulting arrival time is compared with the reference period T_{ref}, where $T_{ref} = N_b * T_b$. When a single photon is detected in a certain time bin, the label of the time bin is recorded as a time tag, converted into binary, and finally output as the final random number. When selecting the length of the time bin, the coincidence count time width as shown in Fig. 2(b) is considered, and the standard deviation is around 500 ps. We select a coincidence count period of four standard deviations as 2 ns to ensure that 99.99% of coincidence events are recorded while avoiding recording noise signals.

In our experiment, the number of time bins is $N_b = 2^4 = 16$, the time bin duration is $T_b = 2$ ns, and the time reference period is $T_{ref} = 32$ ns. In the test, we collected raw data at a rate of 200 kps. Within a reference period, the average number of photons that can be detected is about 0.005, which is far less than 1. It can be considered that the photon arrival time obeys the Poisson distribution. The details of generating quantum random numbers and evaluation of randomness are shown in the supplementary material Note 3.

QRNG from one metalens

We first show how to generate a random sequence from a metalens. Figure 3(a) illustrates how one metalens produces the entangled

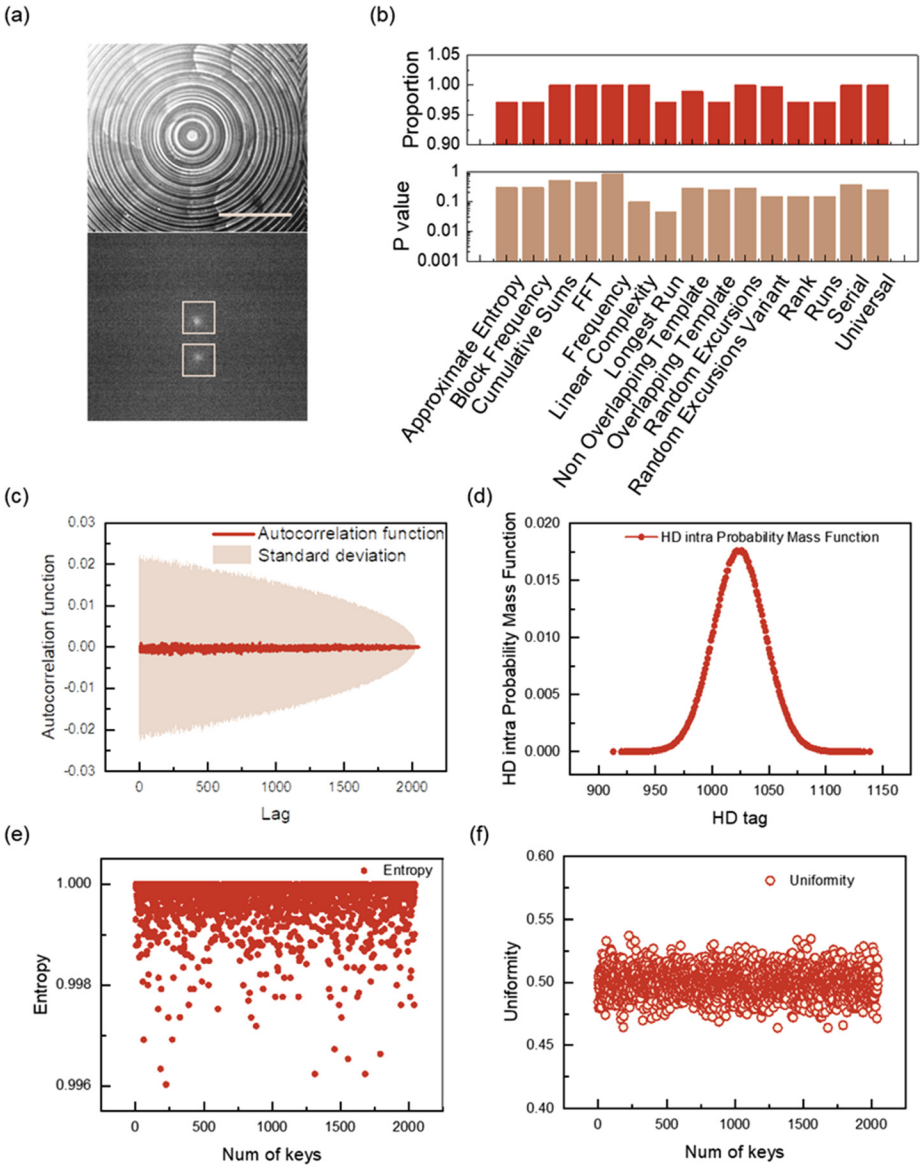

FIG. 3. Entangled photon pairs produced by (a) a metalens, scale bar 50 mm, and (b) the RNG can pass the NIST random test. (c) Strength of Randomness. Autocorrelation function (ACF) plotted as a function of lags. The error bar shows the ACF for all the 2048 keys. (d) Distribution of intra-hamming distance (HD intra) between any two selected streams from the 2048 sequence pairs. (e) Entropy and (f) uniformity for 2048 keys of bit-length 2048 with a mean value.

photon pairs. The SPDC light spots, which represent H and V polarization, respectively, are located on the right side of Fig. 3(a). The NIST statistical test suite is used to assess the random number sequence for 30×10^6. In the test setting, the assessment is set as 1×10^6, and random streams are set as 30. The test's outcome is displayed in Fig. 3(b). As indicated by the result's proportion and P-value, the generated random number can pass every test classification. Since each sequence has a total of about 30 Mbit binary sequences, the amount of data is too large to be processed for further analysis. A random sequence of 2048×2048 bits is chosen sequentially from the quantum random number stream that has been formed. In addition, the random number series was subjected to various randomness testing techniques, such as the Autocorrelation function, the Entropy, the Uniformity of the Sequence, and the intra-hamming distance (HD intra). The ACF displays the relationship between the provided sequence and a delayed replica of the series. Figure 3(c) shows the average and standard deviation of the results of the independent calculation of the ACF for 2048 sequences. Since they are both very close to zero, our random number does not exhibit periodicity, which is a crucial feature of a true random number. The number of flips required to match our random number series to a known sequence is called HD intra. An optimal number of HD intra is half the length of the sequence, as indicated by a binary random number with a 50:50 distribution between 0 and 1. Figure 3(d) displays the HD-intra result for our 2048×2048 sequences. The distribution's mean value is close to 1024, demonstrating the randomness of our values. The degree of uncertainty is called entropy, and its optimum value is 1. Equation (1) allows us to compute the entropy of our sequence,[42]

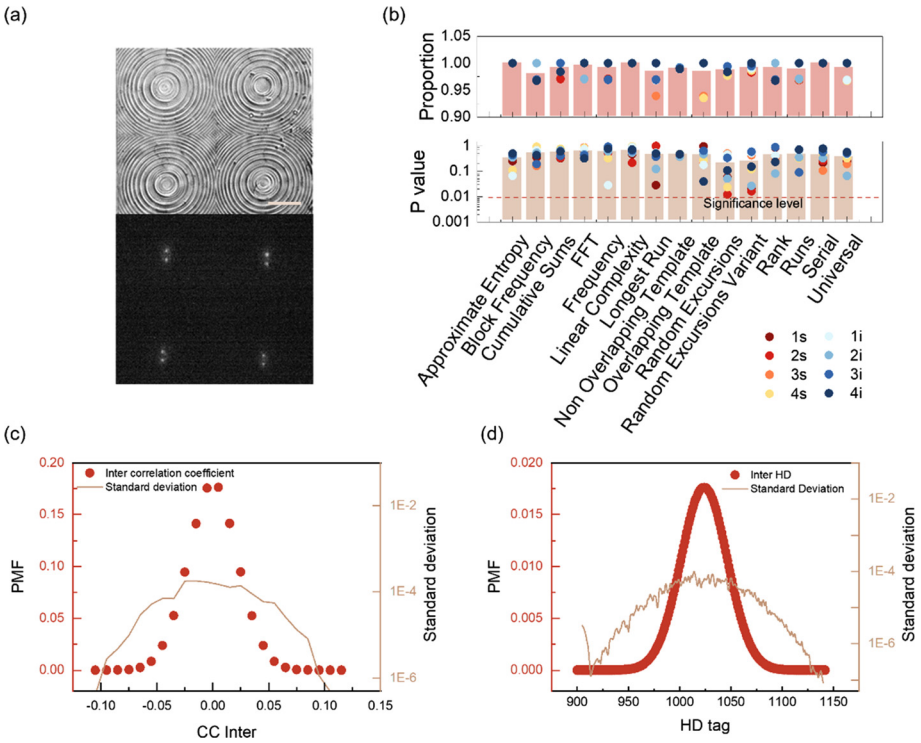

FIG. 4. Photon pairs produced by metalens array and randomness test results. (a) Micrograph of metalens array and the corresponding SPDC light spots, scale bar 50 mm. (b) NIST random test results. (c) Intercorrelation coefficients between sequences. (d) Distribution of inter-hamming distance (HD inter). The standard deviation shows the fluctuation between sequences from metalenses.

$$E = -\left[p\log_2 p + (1-p)\log_2(1-p)\right], \tag{2}$$

where E is the computed entropy, and p and $1-p$ represent the proportion of 0 and 1, respectively, throughout the series. The entropy of around unity, shown in Fig. 2(e), indicates that our data have maximum uncertainty. A uniformity of 0.5 is implied by a good uniformity of binary random numbers, which shows that 0 and 1 have an equal chance of occurring. Our findings, as seen in Fig. 3(f), suggest no bias in our sample since our random numbers are reasonably uniform.

QRNG from metalens array

Here, we present high-dimensional QRNG from a metalens array. Our metalens array is a 10×10 array, but, limited by the number of SPCMs. We only select four metalenses as an example to illustrate the feasibility of high dimensions. Two light spots from the same metalens separately produce a random number. The optical image of the metalens array and the associated light spots are displayed in Fig. 4(a). A quantum random number array is generated by counting the coincidences between each pair of light spots. The results of the NIST randomness test suite are displayed in Fig. 4(b). Every random number stream drawn from various sources passes the test. Figures 4(c) and 4(d) illustrate the random number's physical unclonability. The resemblance between random numbers produced by each metalens is demonstrated using the inter-correlation function and the inter-Hamming distance, which show peaks at 1024 and 0, respectively. This is precisely the optimal value and indicates no similarity between the random number streams obtained from each metalens source.

To obtain secure random numbers, we provide the calculation of the minimum entropy. The minimum entropy is defined as $H_\infty = -\log\left(\mathrm{Max}(P_i)\right)$, where P_i is the detection event probability.[9] All eight random sequences in our experiment are given in Table I. All results are close to 1, which is the minimum entropy of standard uniform distribution with 16 variations. Note that we do not use any randomness extractor here, which is one of the advantages of our approach.

Against GAN attack

Data-driven machine learning demonstrates impressive capabilities in modeling data using probability distributions. It can be utilized to learn the statistical patterns of certain pseudo-random numbers and analyze their vulnerabilities. The key idea behind machine learning is

TABLE I. The calculation of the minimum entropy for eight sequences.

Lens #	1		2		3		4	
Source	s	i	s	i	s	i	s	i
H_{min}	0.9993	0.9991	0.9993	0.9995	0.9995	0.9988	0.9994	0.9992

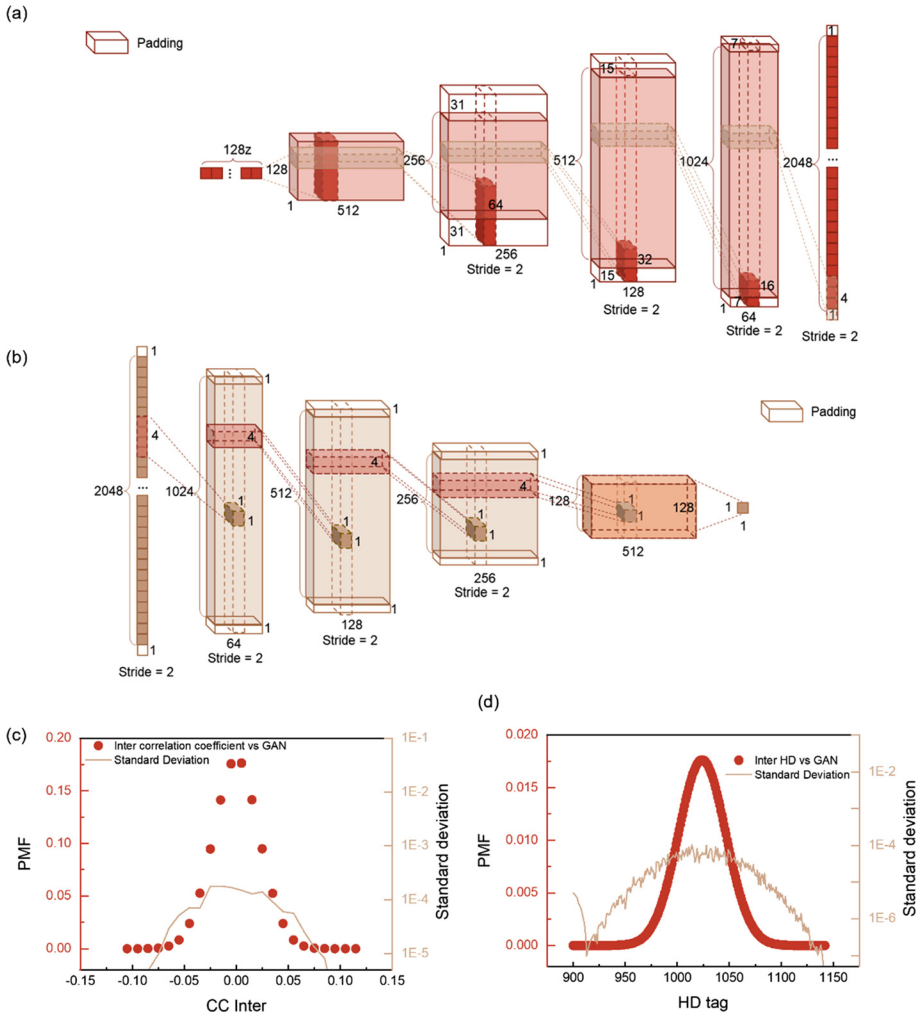

FIG. 5. Resilience to machine learning (ML) attacks. (a) A 1D GAN generator is utilized for pseudo-random number generation. A 128-dimensional uniform distribution Z vector is projected onto a high-level convolutional representation with the same spatial extent but 512 channels. Four fractionally stridden convolutions are further applied to increase the spatial length of the feature maps while reducing the number of channels. Batch normalization and leaky ReLU are applied after each convolution operation, except for the final convolutional layer. The high-dimensional representation is finally transformed into a 2048-dimensional vector output. The final layer employs the sigmoid activation function. (b) A 1D GAN discriminator is used to distinguish between true and pseudo-random numbers. The 2048-dimensional random number vector is projected onto a 128-dimensional feature representation with 512 channels through four 1D convolutions. This high-level feature representation is then converted into a probability value ranging from 0 to 1. Batch normalization is employed after the second, third, and fourth convolutional layers. The activation functions of the former four convolutional layers are leaky ReLU. The final layer employs the sigmoid activation function. No fully connected or pooling layers are present in the generator (a) and discriminator (b), and padding operations are employed to adjust the output size of the convolutional layers. (c) The statistics curve of the CCinter values between the GAN predicted and the experimentally measured streams. (d) The statistics curve of the HD-inter values between the GAN predicted and the experimentally measured keys. The peak value is 1024 for HD-inter values. The standard deviation shows the fluctuation between sequences from the GAN model and metalenses.

to learn statistical distribution in data to perform specific tasks. Real random numbers should have the characteristic of a uniform probability distribution within a given range. Machine learning models cannot learn from QRNG that does not follow any data distribution. A more detailed introduction of the attack of machine learning is shown in the supplementary material Note 4.

Generating a sequence of random numbers with a uniform probability distribution requires that every possible value has an equal probability of being generated. However, neural networks are models based on deterministic algorithms whose outputs are determined by inputs and learned parameters, and they are better at learning complex data distributions. Even though the output sequences obtained by training a neural network appear to be close to a uniform distribution, they are still obtained through a specific calculation process and do not possess true randomness. A neural network can be viewed as a large function approximator. We can train a neural network to generate the output of a pseudo-random number generator (PRNG).[43,44]

To test the randomness of our QRNG, we developed a GAN, a sophisticated machine learning framework, to serve as an adversarial force. GAN aims to learn the statistical characteristics and the corresponding conditional probability distribution model of the dataset and generate fake samples similar to real data. The GAN attacks on well-known PRNGs demonstrated the ability of GANs to learn pseudo-randomness with specific statistical patterns.[44,45] If our QRNG is a PRNG, GAN can learn and crack it. Should the QRNG embody true randomness and eschew any discernible distribution, the GAN will invariably struggle to replicate a similarly random sequence. If the GAN fails to accurately simulate the QRNG output in this struggle, it indicates the true randomness of our QRNG. GAN[46,47] has two modules in the framework: a generator and a discriminator. The observable variable is X, and the target variable is Y. The discriminator establishes the decision boundary to distinguish between real and fake data. The generator learns the statistical model of the joint probability distribution on $X \times Y$ and generates new data using the obtained probability model. The discriminator is a conditional probability model $P(Y \mid x = x)$, which refers to the distribution probability of target Y given the observation x. The generator is a dependent probability model $P(X \mid y = y)$, which is the probability of the distribution of observation X given the target y.

Figure 5(a) demonstrates the architecture of the proposed GAN generator for 1D random number generation. A 128-dimensional noise vector $\mathbb{Z}$ with uniform distribution is projected onto a high-level convolutional representation with the same spatial extent but 512 channels. Each channel contains a specific characteristic response pattern. By increasing the number of feature channels, the network can learn more abstract features at different levels. Four fractionally stridden convolutions are further applied to increase the spatial length of the feature maps while reducing the number of channels. In this up-sampling process, the redundant information is reduced. The features interact with each other and are fused for better integration. To reduce the risks of vanishing and exploding gradient problems, batch normalization[48] and leaky ReLU[49] are applied after each convolution operation, except for the final convolutional layer. The high-dimensional representation is finally transformed into a 2048-dimensional vector output. The final layer employs the sigmoid activation function to regulate the output from 0 to 1. Figure 5(b) displays the framework of the proposed 1D GAN discriminator used for distinguishing between true and false random numbers. The 2048-dimensional random number input vector is projected onto a 128-dimensional feature representation with 512 channels through four 1D convolutions. Such an encoder architecture is designed to extract rich feature representations that can capture important features and patterns in the input data. The encoder can gradually capture higher-level abstract features and develop a comprehensive understanding of the input. This high-level feature representation is then converted into a probability value ranging from 0 to 1.

To compare the randomness produced by the GAN vs that from our metalens array QRNG, we analyzed the inter-correlation function and inter-Hamming distance between the GAN stream and every QRNG stream. The inter-correlation function showed a peak at 0 in Fig. 5(c), while the inter-Hamming distance peaked at 1024 in Fig. 5(d). These optimal values precisely indicate no similarity for the randomness generated from the GAN and the metalens array. Thus, our QRNG output successfully demonstrated decorrelated randomness on par with computationally created random sequences from the GAN. This further verifies the quantum randomness originating from the intrinsic uncertainty of photons emitted by independent metalenses.

DISCUSSION

In summary, we have experimentally realized the generation of high-dimensional quantum random number arrays using metalens arrays. In contrast to previous bulk optics implementations relying on beam splitters, our approach condenses 100 metalenses within a 1 mm^2 footprint. The device has a high-density output with 100 channels per square millimeter, yielding a miniaturized quantum randomness source without post-randomness extraction. There are some alternative physical processes used in QRNGs that can be applied in a similar method to metalens array. Please refer to the supplementary material Note 5, for more details. With the development of on-chip light sources and detectors, it is believed that the size of the completed device will be further reduced. This compact form factor and ease of integration conferred by the metasurface platform signifies advances for miniaturized quantum meta-devices.

MATERIALS AND METHODS

Metalenses design

The metalenses are made up of twelve distinct kinds of nanopillars with variable radii to offer phases between 0 and 330°. The nanopillars are positioned precisely to satisfy the focal lens's phase requirements. The nanopillars' geometrical information is carefully designed by numerical simulation in COMSOL Multiphysics, which has a height of 800 nm and a period of 200 nm. See the supplementary material Note 1, for more details.

Sample fabrication

The sapphire substrate is cleaned by hydrogen baking and coated with a GaN layer using metal-organic chemical vapor deposition (MOCVD), with trimethylgallium (TMGa) and ammonia as precursors. A SiO_2 layer is deposited by plasma-enhanced chemical vapor deposition (PECVD) and patterned by electron-beam lithography (EBL) and Cr evaporation, forming a hard mask for the nanopillars. The GaN layer is etched by inductively coupled plasma reactive ion etching (ICP-RIE) using the SiO_2 mask, creating the metalens arrays. The SiO_2 layer is removed by buffered oxide etch (BOE), leaving the final sample. The detailed fabrication process is shown in supplementary material Fig. S1. The process of fabricating GaN metalenses through reactive ion beam etching is elaborated upon with greater detail in other research articles.[50]

Measurement setup

A 405 nm laser produces entangled photons via spontaneous parametric downconversion in a BBO crystal. The incident beam is focused on the crystal by the metalens array. The emitted light is collected by a 20× objective lens and filtered by a 405 nm notch filter (Semrock NF03-405E-25) and a 475 nm bandpass filter to eliminate the pump light. A polarizing beam splitter (MPBS642, LBTEK) separates the two entangled photons with horizontal and vertical polarization, respectively. A He-Ne laser calibrates the position of the optical elements. A tunable delay line adjusts the path length of the vertically polarized photon. Both horizontally and vertically polarized photons are coupled to multimode fibers and detected by single photon diodes.

An 808 nm bandpass filter (Semrock LL01-808-25) removes residual pump light and ambient light. The detected photons are recorded by a time-to-digital converter for further analysis. A qCMOS camera verifies the alignment of the metalens and the BBO. See the supplementary material Note 2, for more details.

GAN model training

The two modules play with each other in a zero-sum game where one's gain is the other's loss. Given a random number sequence that may be either a real sequence from a real physical QRNG or a fake sequence generated by the generator, the discriminator should determine whether the sequence is real or fake. On the contrary, the generator tries to generate a fake sequence that can fool the discriminator. In training, two modules try to complete their conditional probability model. After training, the generator masters the distribution of real data to generate new random number sequences.

We trained the GAN on the homemade random number dataset collected from our QRNG. There are about 1.65×10^8 binary numbers in this dataset. Using binary data directly as the output of a GAN generator can lead to difficult training and convergence issues. Binary data are discrete and highly discontinuous, with huge gaps from the continuous output of the generator. The GAN generator aims to generate realistic data samples that can fool the GAN discriminator. During training, the generator adjusts its parameters via backpropagation to minimize the loss function. However, propagation and adjustment of the gradients become complicated when the output target of the generator is discrete binary data. Even if the Sigmoid activation function[51] is employed, training will face the challenge of non-convergence. The Sigmoid function can map the entire real number axis to the interval from 0 to 1. However, the gradients at the outputs 0 and 1 are extremely small, close to zero. The corresponding inputs are extremely large (negative infinity and positive infinity). These will quickly lead to vanishing gradient and exploding gradient problems. Therefore, we encode the raw binary data into the signed integer with 8 bits. Every 8 binary number is converted to a signed integer, ranging from -128 to 127. The signed integers are further normalized into the floating point numbers ranging from 0 to 1, as shown in the following equation:

$$S_f = \frac{S_i + 128}{255}, \tag{3}$$

where S_f is the normalized floating number, S_i is the signed integer. After the normalization, the dataset comprises 8000 training samples and 2000 test samples. Each sample contains 2048 normalized floating point numbers.

Batch normalization is employed after the second, third, and fourth convolutional layers. Similar to the generator, the activation functions of the former four convolutional layers are leaky ReLU. A probability output of 0 indicates a fake sample, while a probability output of 1 signifies a real sample. There are no fully connected or pooling layers in the generator and discriminator. The padding operations are employed to adjust the output size of the convolutional layers.

We use the binary cross entropy (BCE)[52] to construct the loss functions of the generator and the discriminator, which is defined as

$$\ell(x, y) = \frac{1}{N}\sum_{n=1}^{N} -\left[y_n \cdot logx_n + (1 - y_n) \cdot \log(1 - x_n)\right], \tag{4}$$

where $\mathcal{X} = \{x_n\}_{n=1}^{N}$ is the batch of prediction probabilities, $\mathcal{Y} = \{y_n\}_{n=1}^{N}$ is the batch of the labels (1 for real and 0 for fake), and N is the batch size. The training loop updates the discriminator first. We select a batch of real samples from the training set, forward pass D, and calculate the loss $\ell(D_1(x), 1)$. A batch of fake samples is generated by the current generator. The loss for the fake batch is calculated as $\ell\big(D_1(G(\mathcal{Z})), 0\big)$. The loss function of the discriminator is calculated as the sum of the losses for both real and fake batches and the regularization,[53] as shown in the following equation:

$$\begin{aligned} Loss_D &= \ell(D_1(x), 1) + \ell\big(D_1(G(\mathcal{Z})), 0\big) + \lambda_D \sum_j wD_j^2 \\ &= \frac{1}{N}\sum_{n=1}^{N} -\left[\log(D_1(x)) + \log(1 - D_1(G(\mathcal{Z})))\right] \\ &\quad + \lambda_D \sum_j wD_j^2, \end{aligned} \tag{5}$$

where $D_1(x)$ is the output probability of the discriminator when its input is the real sample x from the training set, $\mathcal{Z}$ is the input vector of the generator, $G(\mathcal{Z})$ is the output vector of the generator, $D_1(G(\mathcal{Z}))$ is the output probability of the discriminator when its input is the fake sample $G(\mathcal{Z})$, λ_D is the regularization coefficient, and $\mathcal{WD} = \{wD_j\}_{j=1}^{J}$ are the trainable weights in the network. To minimize $Loss_D$, $D_1(x)$ should be 1, and $D_1(G(\mathcal{Z}))$ should be 0, which means the discriminator can distinguish the real and fake samples correctly. With $Loss_D$, the training parameters in the discriminator are updated by the gradient backpropagation algorithm. In the following training of the generator, the loss function of the generator is calculated as

$$\begin{aligned} Loss_G &= \ell\big(D_2(G(\mathcal{Z})), 1\big) + \lambda_G \sum_j wG_j^2 \\ &= \frac{1}{N}\sum_{n=1}^{N} -\log(D_2(G(\mathcal{Z}))) + \lambda_G \sum_j wG_j^2, \end{aligned} \tag{6}$$

where $D_2(G(\mathcal{Z}))$ is the output of the discriminator after its update, λ_G is the regularization coefficient, and $\mathcal{WG} = \{wG_j\}_{j=1}^{J}$ are the trainable weights in the network. The purpose of the generator is to fool the discriminator. Training of the generator aims to minimize the $Loss_G$, which computes the deviations between the prediction $D(G(\mathcal{Z}))$ and real label $y = 1$.

The generator transforms a low-dimensional random vector into a high-dimensional sample that closely resembles a real sample. Typically, the generator exhibits greater complexity and possesses more trainable parameters than the discriminator. Our discriminator comprises approximately 755 thousand trainable parameters, whereas the generator consists of around 17×10^6 trainable parameters. The generator's parameters are approximately 24 times that of the discriminator. Establishing a coordinated and balanced training process between the generator and discriminator is crucial. Optimizing the generator is a more challenging task. The initial learning rate of GAN should be relatively small. An excessive learning rate may lead to instability in the training process, manifesting as an imbalance in the game between the generator and the discriminator, making it difficult for GAN to reach a convergence state. After hyperparameter attempts, the initial learning rates for the discriminator and the generator are 2×10^{-6} and 1.3×10^{-6}, respectively. As the training progresses, the

discriminators, which have a relatively more straightforward task, tend to improve steadily. To ensure a balanced interplay between the two components, it is beneficial to maintain a higher learning rate for the generator during the later stages of training. This can be achieved by setting the learning rate decay factor of 0.95 for the generator and 0.9 for the discriminator every 200 epochs. This approach facilitates faster learning and adaptation of the generator to the feedback provided by the discriminator. To avoid overfitting, the regularization coefficients for the discriminator and generator are 0.001 and 0.023, respectively.

We employed the Adam Optimizer ($\beta = 0.999$ for discriminator and generator). The batch size was 16 on a Nvidia GeForce RTX 3090 GPU. After 2000 epochs (1 000 000 iterations) of training, the final $D_1(x)$ and $D(G(\mathcal{Z}))$ converged to approximately 0.5, realizing Nash equilibrium.[54] When a GAN converges, the output of the generator becomes consistent with the distribution of real samples, and the discriminator cannot accurately distinguish them. At this time, the discriminator's judgment of real samples and generated samples is almost random, so the probability of output is close to 0.5.

Device modeling and assumptions

We first consider the randomness of the spontaneous emission of light. The emission of stimulated photons happens randomly as transitions take place from an upper energy level to a lower energy level, with a transition probability denoted as p_0.Then, the randomness of the spontaneous parametric downconversion (SPDC) process is considered. Photons from the excited light interact with a nonlinear crystal, resulting in the generation of SPDC with a probability denoted as p_1, probability of no generation is $1 - p_1$. The total SPDC generation probability is

$$p = p_0 p_1. \tag{7}$$

Consider a continuous-wave (CW) laser over a specific time interval, where the upper energy level is populated with N charge carriers, k of which transition to the lower level to generate excited light. This process is modeled by a binomial distribution expression,

$$P(X = k) = \binom{N}{k} p^k (1-p)^{N-k}. \tag{8}$$

Continuing the analysis with the assumption of a sufficiently large N and a sufficiently small p, the limit is taken, yielding the generation times (n) of SPDC as a Poisson distribution with a mean generation time of parameter $\lambda = \eta N p$, where η is the total detection efficiency. According to energy conservation, when a photon falls, it can only produce at most one SPDC photon pair. Therefore, the generation times of SPDC are the pair numbers of SPDC generation,

$$P(\widehat{K} = k) = \frac{\lambda^k}{k!} e^{-\lambda}. \tag{9}$$

The time difference between photon arrival time and time reference is approximately uniform distribution, which we collect to generate QRNG. We have quantitatively evaluated the randomness by the minimum entropy in the main text. We analyze our model in the following. There are several assumptions for our model.

1. Dark count: The dark counts can be disregarded compared to the count rate, as the dark counts for the SPCM amount to only 100 counts per second.
2. Detector and metalens efficiency: Considering the efficiency of two detectors, η_A and η_B, and the efficiency of the metalens η_M, the total detection efficiency is $\eta = \eta_A \eta_B \eta_M$. Thus, we can refine the probability distribution of photon pairs generated through SPDC.
3. Dead time: The SPCM has a dead time of τ_d. Dead time is a period that a detector is inactive after detection that does not affect the randomness of the raw data.[22]
4. Multiphoton pair emission from the nonlinear crystal: When k photon pairs appear in a period, every k detection event will be announced for an ideal detector without dead time. However, in the experiment, only the first detection event is recorded as the raw data. Therefore, for a detection event, the conditional probability of getting the result $\widehat{n} = i$ given that k photons appear in a period,

$$P(\widehat{n} = i|k) = \left(1 - \frac{i-1}{N_b}\right)^k - \left(1 - \frac{i}{N_b}\right)^k, \tag{10}$$

where $i = 1, 2, \ldots, N_b$. Considering the probability distributions of the pair number of SPDC generation Equation (M7), the maximum probability occurs when the photon drops into the first time bin,

$$\begin{aligned} P(\widehat{n} = 1) &= \frac{1}{1 - e^{-\lambda}} \sum_{k=1}^{\infty} P(\widehat{n} = 1|k) P(\widehat{K} = k) \\ &= \frac{1}{1 - e^{-\lambda}} \sum_{k=1}^{\infty} \left(1 - \left(1 - \frac{1}{N_b}\right)^k\right) \frac{\lambda^k}{k!} e^{-\lambda}. \end{aligned} \tag{11}$$

Expanding the first term within the summation through a series expansion, when $N_b > 1$,

$$\leq \frac{1}{1 - e^{-\lambda}} \sum_{k=1}^{\infty} \frac{k}{N_b} \frac{\lambda^k}{k!} e^{-\lambda} = \frac{\lambda}{N_b (1 - e^{-\lambda})}. \tag{12}$$

The TDC exhibits a time resolution of 16 ps, rendering its impact negligible compared to the SPCM. After the photon arrives at the SPCM, the output signal, denoted by the function f, is influenced by the jitter-rising edge of the square wave output that conforms to a Gaussian distribution. Specifically, the arrival time of the photon at the SPCM is represented as t_0, while the time indicated by the rising edge of the SPCM output square wave is denoted as $t_1 = f(t_0) = t_0 + n$, n is the jitter caused by SPCM, obeying the normal distribution, $n \sim N(\mu, \sigma^2)$. We denote the two generated photons with subscripts s and i. The true value of the time difference between these two photons is represented by dt. The measured value of the time difference is $t = t_{1s} - t_{1i} = f(t_{1s}) - f(t_{1i}) = dt + n_s - n_i$.

The expected full width at half maximum (FWHM) of the time difference statistics between two SPDC photons is influenced by instrumental jitter in the measurement values $n_{total} = n_s - n_i$, obeying $n_{total} \sim N(0, 2\sigma^2)$. Finally, we have FWHM $= \sqrt{2}\sigma \approx 495\,ps$, which is very close to the experimental fitting value 500 ps.

SUPPLEMENTARY MATERIAL

See the supplementary material for details on the design, processing, and characterization of metalens array; experimental details of random number generation; details on methods used for randomness

extraction and verification; and details on generative adversarial networks.

ACKNOWLEDGMENTS

This work is supported by the University Grants Committee/Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. AoE/P-502/20, CRF Project: C1015-21E; C5031-22G, GRF Project: CityU15303521; CityU11305223; CityU11310522; CityU11300123, and Germany/Hong Kong Joint Research Scheme: G-CityU 101/22), City University of Hong Kong (Project No. 9380131, 9610628, and 7005867), and the National Natural Science Foundation of China (Project No. 62375232).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Yubin Fan, Shufan Chen, and Xiaoyuan Liu contributed equally to this work.

Yubin Fan: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Project administration (equal); Resources (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Shufan Chen:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Xiaoyuan Liu:** Data curation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **Xiaoyu Che:** Data curation (equal); Formal analysis (equal); Visualization (equal). **Xiaodong Qiu:** Conceptualization (equal); Methodology (equal); Resources (equal); Validation (equal). **Mu Ku Chen:** Funding acquisition (equal); Resources (equal). **Din Ping Tsai:** Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Supervision (equal); Validation (equal); Writing – review & editing (equal).

DATA AVAILABILITY

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

REFERENCES

[1]T. F. Blauth, O. J. Gstrein, and A. Zwitter, IEEE Access **10**, 77110–77122 (2022).
[2]M. Herrero-Collantes and J. C. Garcia-Escartin, Rev. Mod. Phys. **89**(1), 015004 (2017).
[3]S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. L. Pereira, M. Razavi, J. S. Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden, Adv. Opt. Photonics **12**(4), 1012–1236 (2020).
[4]V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, Rev. Mod. Phys. **81**(3), 1301–1350 (2009).
[5]F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, Rev. Mod. Phys. **92**(2), 025002 (2020).
[6]J. Yin, Y. Cao, Y. H. Li, S. K. Liao, L. Zhang, J. G. Ren, W. Q. Cai, W. Y. Liu, B. Li, H. Dai, G. B. Li, Q. M. Lu, Y. H. Gong, Y. Xu, S. L. Li, F. Z. Li, Y. Y. Yin, Z. Q. Jiang, M. Li, J. J. Jia, G. Ren, D. He, Y. L. Zhou, X. X. Zhang, N. Wang, X. Chang, Z. C. Zhu, N. L. Liu, Y. A. Chen, C. Y. Lu, R. Shu, C. Z. Peng, J. Y. Wang, and J. W. Pan, Science **356**(6343), 1140–1144 (2017).
[7]C. Wu, B. Bai, Y. Liu, X. Zhang, M. Yang, Y. Cao, J. Wang, S. Zhang, H. Zhou, X. Shi, X. Ma, J. G. Ren, J. Zhang, C. Z. Peng, J. Fan, Q. Zhang, and J. W. Pan, Phys. Rev. Lett. **118**(14), 140402 (2017).
[8]J.-N. Zhang, R. Yang, X. Li, C.-W. Sun, Y.-C. Liu, Y. Wei, J.-C. Duan, Z. Xie, Y.-X. Gong, and S.-N. Zhu, Adv. Photonics **5**(03), 036003 (2023).
[9]Y.-Q. Nie, H.-F. Zhang, Z. Zhang, J. Wang, X. Ma, J. Zhang, and J.-W. Pan, Appl. Phys. Lett. **104**(5), 051110 (2014).
[10]M. Eaton, A. Hossameldin, R. J. Birrittella, P. M. Alsing, C. C. Gerry, H. Dong, C. Cuevas, and O. Pfister, Nat. Photonics **17**(1), 106–111 (2022).
[11]W. Luo, L. Cao, Y. Shi, L. Wan, H. Zhang, S. Li, G. Chen, Y. Li, S. Li, Y. Wang, S. Sun, M. F. Karim, H. Cai, L. C. Kwek, and A. Q. Liu, Light **12**(1), 175 (2023).
[12]G. Zhang, J. Y. Haw, H. Cai, F. Xu, S. M. Assad, J. F. Fitzsimons, X. Zhou, Y. Zhang, S. Yu, J. Wu, W. Ser, L. C. Kwek, and A. Q. Liu, Nat. Photonics **13**(12), 839–842 (2019).
[13]J. Wang, F. Sciarrino, A. Laing, and M. G. Thompson, Nat. Photonics **14**(5), 273–284 (2020).
[14]E. Pelucchi, G. Fagas, I. Aharonovich, D. Englund, E. Figueroa, Q. Gong, H. Hannes, J. Liu, C.-Y. Lu, and N. Matsuda, Nat. Rev. Phys. **4**(3), 194–208 (2022).
[15]Y. Fan, H. Liang, J. Li, D. P. Tsai, and S. Zhang, ACS Photonics **9**(9), 2872–2890 (2022).
[16]J. Yao, J.-Y. Ou, V. Savinov, M. K. Chen, H. Y. Kuo, N. I. Zheludev, and D. P. Tsai, PhotoniX **3**(1), 23 (2022).
[17]W. C. Wong, K. M. Lau, H. Liang, T. K. Yung, B. Zeng, and J. Li, Phys. Rev. A **106**(1), 013503 (2022).
[18]Y. Wang, Y. Fan, X. Zhang, H. Tang, Q. Song, J. Han, and S. Xiao, ACS Nano **15**(4), 7386–7391 (2021).
[19]Y. Fan, P. Tonkaev, Y. Wang, Q. Song, J. Han, S. V. Makarov, Y. Kivshar, and S. Xiao, Nano Lett. **21**(17), 7191–7197 (2021).
[20]X. Wang, T. Sentz, S. Bharadwaj, S. K. Ray, Y. Wang, D. Jiao, L. Qi, and Z. Jacob, Sci. Adv. **9**(4), eade4203 (2023).
[21]X. Wang, Z. Yang, F. Bao, T. Sentz, and Z. Jacob, Optica **11**(1), 73–80 (2024).
[22]J. B. Khurgin and G. Sun, Nat. Photonics **8**(6), 468–473 (2014).
[23]J. Ren, H. Lin, X. Zheng, W. Lei, D. Liu, T. Ren, P. Wang, and B. Jia, Opto-Electron. Sci. **1**(6), 210013 (2022).
[24]G. Yinghui, P. Mingbo, Z. Fei, X. Mingfeng, L. Xiong, M. Xiaoliang, and L. Xiangang, Photonics Insights **1**(1), R03 (2022).
[25]E. Choi, G. Kim, J. Yun, Y. Jeon, J. Rho, and S. H. Baek, Nat. Photonics **18**, 848–855 (2024).
[26]J. Kim, H. Kim, H. Kang, W. Kim, Y. Chen, J. Choi, H. Lee, and J. Rho, Nat. Food **5**(4), 715 (2024).
[27]J. Kim, Y. Kim, W. Kim, D. K. Oh, D. Kang, J. Seong, J. W. Shin, D. Go, C. Park, H. Song, J. An, H. Lee, and J. Rho, Mater. Today **73**, 9–15 (2024).
[28]J. Kim, J. Seong, W. Kim, G. Y. Lee, S. Kim, H. Kim, S. W. Moon, D. K. Oh, Y. Yang, J. Park, J. Jang, Y. Kim, M. Jeong, C. Park, H. Choi, G. Jeon, K. I. Lee, D. H. Yoon, N. Park, B. Lee, H. Lee, and J. Rho, Nat. Mater. **22**(4), 474 (2023).
[29]S. Li, Y. Fang, and J. Wang, Opto-Electron. Sci. **3**(7), 240011 (2024).
[30]S. W. Moon, J. Kim, C. Park, W. Kim, Y. Yang, J. Kim, S. Lee, M. Choi, H. Sung, J. Park, H. Song, H. Lee, and J. Rho, Laser Photonics Rev. **18**(4), 2300929 (2024).
[31]L. Li, Z. Liu, X. Ren, S. Wang, V. C. Su, M. K. Chen, C. H. Chu, H. Y. Kuo, B. Liu, W. Zang, G. Guo, L. Zhang, Z. Wang, S. Zhu, and D. P. Tsai, Science **368**(6498), 1487–1490 (2020).
[32]Q. Li, W. Bao, Z. Nie, Y. Xia, Y. Xue, Y. Wang, S. Yang, and X. Zhang, Nat. Photonics **15**(4), 267–271 (2021).
[33]C. Altuzarra, A. Lyons, G. Yuan, C. Simpson, T. Roger, J. S. Ben-Benjamin, and D. Faccio, Phys. Rev. A **99**(2), 020101 (2019).
[34]K. Wang, J. G. Titchener, S. S. Kruk, L. Xu, H. P. Chung, M. Parry, I. I. Kravchenko, Y. H. Chen, A. S. Solntsev, Y. S. Kivshar, D. N. Neshev, and A. A. Sukhorukov, Science **361**(6407), 1104–1108 (2018).
[35]J. Wang, S. Paesani, Y. Ding, R. Santagati, P. Skrzypczyk, A. Salavrakos, J. Tura, R. Augusiak, L. Mancinska, D. Bacco, D. Bonneau, J. W. Silverstone, Q. Gong,

A. Acin, K. Rottwitt, L. K. Oxenlowe, J. L. O'Brien, A. Laing, and M. G. Thompson, Science **360**(6386), 285–291 (2018).
[36]Q. Guo, X. Z. Qi, L. Zhang, M. Gao, S. Hu, W. Zhou, W. Zang, X. Zhao, J. Wang, B. Yan, M. Xu, Y. K. Wu, G. Eda, Z. Xiao, S. A. Yang, H. Gou, Y. P. Feng, G. C. Guo, W. Zhou, X. F. Ren, C. W. Qiu, S. J. Pennycook, and A. T. S. Wee, Nature **613**(7942), 53–59 (2023).
[37]Y. Fan, H. Liang, Y. Wang, S. Chen, F. Lai, M. Ku Chen, S. Xiao, J. Li, and D. P. Tsai, Adv. Photonics Nexus **3**(1), 016011 (2024).
[38]G. Sun, R. A. Soref, J. B. Khurgin, S. Q. Yu, and G. E. Chang, Opt. Express **30**(23), 42385–42393 (2022).
[39]K.-C. Shen, Y.-T. Huang, T. L. Chung, M. L. Tseng, W.-Y. Tsai, G. Sun, and D. P. Tsai, Phys. Rev. Appl. **12**(6), 064056 (2019).
[40]J. Liu, M. Q. Shi, Z. Chen, S. M. Wang, Z. L. Wang, and S. N. Zhu, Opto-Electron. Adv. **4**(9), 200092 (2021).
[41]D. Bunandar, Z. Zhang, J. H. Shapiro, and D. R. Englund, Phys. Rev. A **91**(2), 022336 (2015).
[42]H. Ravichandran, D. Sen, A. Wali, T. F. Schranghamer, N. Trainor, J. M. Redwing, B. Ray, and S. Das, ACS Nano **17**(17), 16817–16826 (2023).
[43]K. Tirdad and A. Sadeghian, in *Annual Meeting of the North American Fuzzy Information Processing Society* (IEEE, 2010), pp. 1–6.
[44]M. De Bernardi, M. Khouzani, and P. Malacaria, in *ECML PKDD 2018* (Springer, 2019), pp. 191–200.
[45]K. Okada, K. Endo, K. Yasuoka, and S. Kurabayashi, PLOS One **18**(6), e0287025 (2023).
[46]I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Advances in Neural Information Processing Systems* (NeurIPS Proceedings, 2014), Vol. 27.
[47]A. Radford, L. Metz, and S. Chintala, arXiv:1511.06434 (2015).
[48]N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, *Advances in Neural Information Processing Systems* (NeurIPS Proceedings, 2018), Vol. 31.
[49]A. L. Maas, A. Y. Hannun, and A. Y. Ng, in Presented at the Proceedings of International Conference on Machine Learning (2013).
[50]B. Abasahl, C. Santschi, T. V. Raziman, and O. J. F. Martin, Nanotechnology **32**(47), 475202 (2021).
[51]M. Rezaeian Zadeh, S. Amin, D. Khalili, and V. P. Singh, Water Resour. Manage. **24**, 2673–2688 (2010).
[52]U. Ruby and V. Yendapalli, Int. J. Adv. Trends Comput. Sci. Eng. **9**(10), 5393–5397 (2020).
[53]T. Van Laarhoven, arXiv:1706.05350 (2017).
[54]C. A. Holt and A. E. Roth, Proc. Nat. Acad. Sci. **101**(12), 3999–4002 (2004).