Supplementary Information for

Harnessing Optoelectronic Noises in a Hybrid Photonic Generative Adversarial Network (GAN)

Changming Wu[1], Xiaoxuan Yang[2], Heshan Yu[3], Ruoming Peng[1], Ichiro Takeuchi[3], Yiran Chen[2] and Mo Li[1,4]

[1]*Department of Electrical and Computer Engineering, University of Washington, Seattle, WA 98195, USA*

[2]*Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA*

[3]*Department of Materials Science and Engineering, University of Maryland, College Park, MD 20742, USA*

[4]*Department of Physics, University of Washington, Seattle, WA 98195, USA*

I. Theory and process flow for O/E random number generator (RNG)

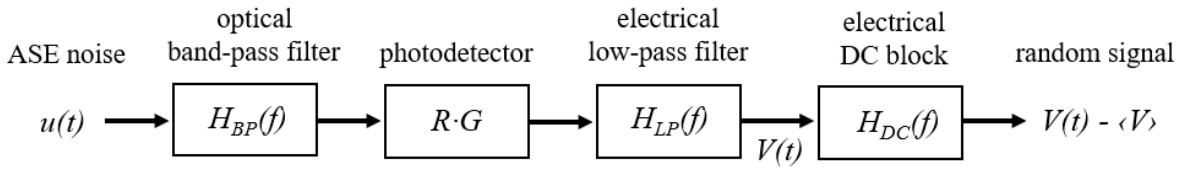

Fig. S1 The block diagram showing the key component and the process flow in the random number generation procedure using the ASE noise as the input.

In this work, we followed the same method developed by Williams et al.[1] to generate random signals from the ASE noise. Fig. S1 shows the process flow to generate random signals. The input signal $u(t)$ is the optical random noise produced from amplified spontaneous emission (ASE) and has a broadband power spectral density (PSD) $S_{in}(f)$, as shown in Fig. 2a in the main text. The input is first spectrally filtered after passing an optical band-pass filter with the frequency response $H_{BP}(f)$. Due to the narrow bandwidth of the band-pass filter B_{BP}, the PSD within the bandwidth is approximated as a constant S_{in} and the PSD after the band-pass filter becomes $S_{in}|H_{BP}(f)|^2$. The filtered optical signal is then detected by a square-law photodetector and further passes an electrical low-pass filter with the frequency response $H_{LP}(f)$ and the bandwidth B_{LP}, generating noisy baseband electrical voltage signals from the beating between different optical frequency

components, which are referred to as "ASE-ASE beat noises". In practice, the photodetector plays the role of both the power meter and the low-pass filter. Here, we assume the power responsivity and the gain of the photodetector are constants, R (in V/mW) and G, respectively. The frequency responses of the band-pass and low-pass filters are Gaussian following:

$$|H_{BP}(f)|^2 = \exp\left[-(4ln2)\frac{(f-f_0)^2}{B_{BP}^2}\right], \qquad |H_{LP}(f)|^2 = \exp\left[-(ln2)\frac{f^2}{B_{LP}^2}\right].$$

where f_0 is the center frequency of the band-pass filter. As a result, PSD of the voltage noise S_{noise} obtained after the electrical DC block is given by the multiplication between total integration of optical noise power and the photodetector gain and responsitivity, is a Gaussian:

$$\begin{aligned} S_{noise}(f) &= R^2G^2S_{in}^2|H_{LP}(f)|^2\int|H_{BP}(f')H_{BP}(f'+f)|^2df' \\ &= R^2G^2S_{in}^2B_{BP}\sqrt{\frac{\pi}{8ln2}}\exp\left[-(ln2)\left(\frac{1}{B_{LP}^2}+\frac{2}{B_{BP}^2}\right)f^2\right]. \end{aligned}$$

The corresponding voltage variance at the final output thus is given by:

$$\begin{aligned} \sigma_{noise}^2 &= \int S_{noise}(f)df = R^2G^2S_{in}^2\int\int|H_{LP}(f)H_{BP}(f')H_{BP}(f'+f)|^2df'df \\ &= R^2G^2S_{in}^2B_{BP}\sqrt{\frac{\pi}{4ln2}}(1+\frac{B_{BP}^2}{2B_{LP}^2})^{-1/2}. \end{aligned}$$

In practice, we control the mean power of the ASE noise (DC component), $S_{in}H_{LP}(0)\int|H_{BP}(f)|^2df = S_{in}B_{BP}\sqrt{\frac{\pi}{4ln2}}$, and make sure that it will not saturate the photodetector.

II. Measurement setup for photonic convolutional tensor core

The experimental setup used to carry out the convolution operations with a 2×2 matrix (see Fig.1 of the main text) is shown in Fig. S2. We follow the same process of operating the photonic network as described in Wu *et.al*[2]. The input vector is encoded as the temporal modulated optical signal from four different wavelength channels. Another laser connected to a 1×4 optical switch is

used to selectively set the mode contrast of individual PMMC, which represents a single kernel weight element. The MVM operation is reduced to the mode contrast measurement. The resulting transmitted power of TE_0 and TE_1 modes are summed incoherently using photodetectors. Their difference is calculated electronically and used in post-processing steps. The TE_0 mode output coming from all the PMMCs is combined using on-chip Y-junctions, while the TE_1 mode output power is combined off-chip because the same ports are used to program the PMMCs optically. Because combining four incoherent sources using Y-junctions will inherently reduce the power by a factor of 4, we rescaled the measured TE_0 mode power by this factor when calculating the power differences between the two modes.

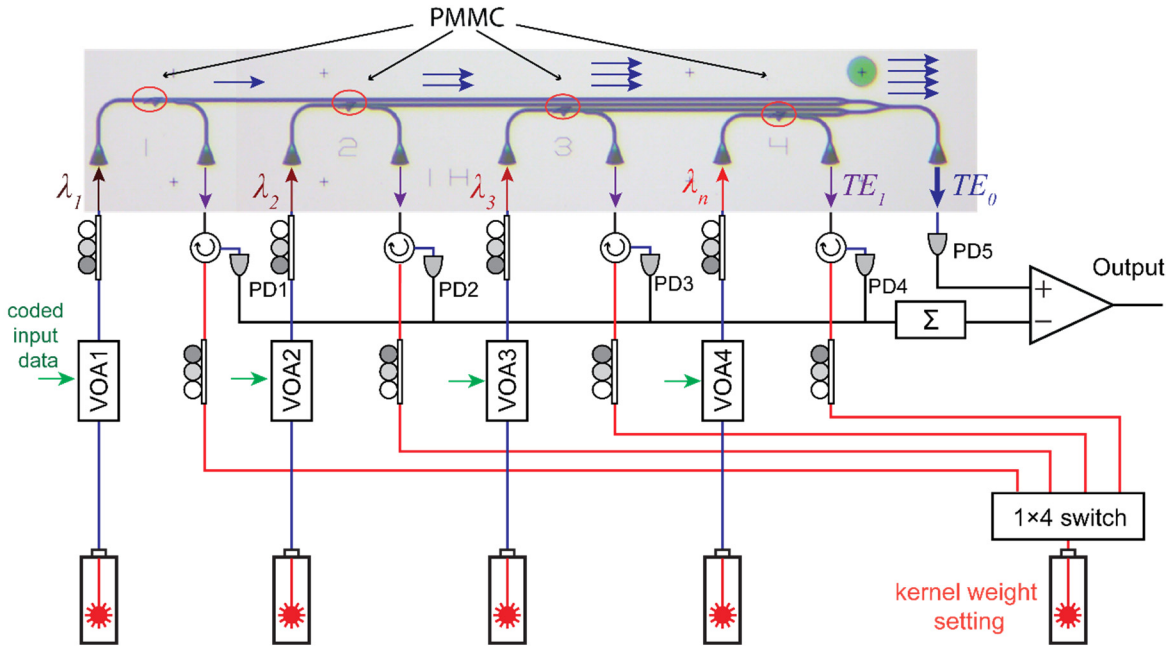

Fig. S2 Experimental setup for convolutional MVM operation. The input vector is encoded as the temporal modulated optical signal from four different wavelength channels and sent into four PMMCs (solid blue lines). Another laser connected to a 1×4 optical switch is used to selectively set the mode contrast of individual PMMC, which represents a single kernel weight element (solid red lines). The MVM operation is reduced to the mode contrast measurement. The total TE_0 mode power is collected on-chip using Y junctions while the TE_1 mode power is collected off-chip by an external electrical circuit (solid black lines).

III. MVM error analysis for PMMC based photonic tensor core

The key to generating high quality handwritten number images is to perform accurate matrix-vector multiplication (MVM) operations, $\mathbf{Y}^l=\mathbf{W}^l\cdot\mathbf{X}^l$, where $\mathbf{Y}^l$ is the output matrix, $\mathbf{W}^l$ is the kernel

matrix and $\mathbf{X}^l$ is the input vector of the layer l. In practice the noise is inevitably introduced through non-ideal pieces of equipment, leading to errors on both input vector $\mathbf{X}^l$ and the kernel matrix $\mathbf{W}^l$, the realistic MVM operation will give the results $\mathbf{Y}^{l\prime}=\mathbf{Y}^l+\mathbf{\Delta Y}^l=(\mathbf{W}^l+\mathbf{\Delta W}^l)\cdot(\mathbf{X}^l+\mathbf{\Delta X}^l)\approx \mathbf{W}^l\cdot\mathbf{X}^l+\mathbf{\Delta W}^l\cdot\mathbf{X}^l+\mathbf{W}^l\cdot\mathbf{\Delta X}^l$, where $\mathbf{\Delta Y}^l$, $\mathbf{\Delta W}^l$, $\mathbf{\Delta X}^l$ are the errors for the corresponding elements. $\mathbf{\Delta X}^l$ is mainly caused by the inaccurate response of the EOM at the input. $\mathbf{\Delta W}^l$ is caused by mode contrast setting error, $\Delta\Gamma^l$, which includes short-term mode contrast setting inaccuracy $\delta\Gamma$ (write noise) as well as the long-term measurement fluctuations (read noise) such as the drift of the measurement setup over time. The ratio between the two error terms contribute to the element Y^l_i of layer l during the MVM calculation is estimated by:

$$\frac{(\mathbf{\Delta W^l}\cdot\mathbf{X^l})_i}{(\mathbf{W^l}\cdot\mathbf{\Delta X^l})_i}=\frac{\sum\Delta w^l{}_{ij}x_j^l}{\sum w^l{}_{ij}\Delta x_j^l}\approx\frac{\langle\frac{|\Delta w^l|}{|w^l|}\rangle}{\langle\frac{|\Delta x^l|}{|x^l|}\rangle}\approx\frac{\frac{\Delta\Gamma^l}{\langle|\Gamma^l|\rangle}}{\frac{\delta|x^l|}{\langle|x^l|\rangle}},$$

where the $\Delta\Gamma^l$ and $\delta|x^l|$ are the standard deviations (STD) of mode contrast setting error and the input error, $\langle|\Gamma^l|\rangle$ and $\langle|x^l|\rangle$ are the mean values of the mode contrast setting error and the input error. Take the first layer as an example, the $\langle x^l\rangle$ and $\delta|x^l|$ are 0.1723 and 8×10^{-4} respectively (see Fig.S3). The $\langle|\Gamma^l|\rangle$ and $\Delta\Gamma^l$ are 0.4236 and 0.05 for weight setting error (0.007 for short-term write noise $\delta\Gamma$). The ratio between the two error terms is 25.42 (4.07 only short-term write noise considered), thus the input setting error will not be the main noise source in our photonic GAN.

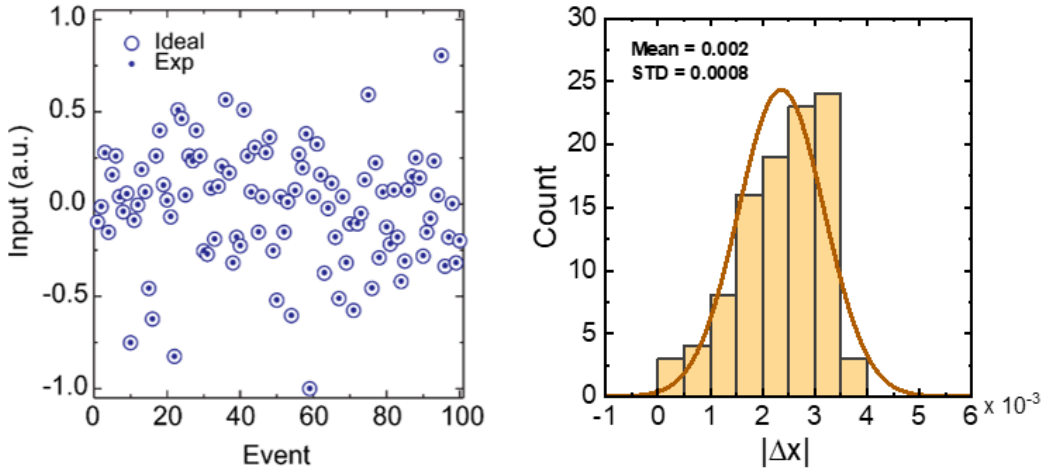

Fig. S3 a. An encoded optical signal trace is used as the input in the first layer obtained from measurement with its corresponding ideal value. **b.** The histogram of the input signal error is defined as $|\Delta x|=|x_{experiment}-x_{ideal}|$. The standard deviation is only 8×10^{-4}, much smaller than the STD of short-time mode contrast setting error $\delta\Gamma$ (~0.8%) and the overall contrast setting error $\Delta\Gamma^l$ (~5%).

IV. Noise-Aware training Method for GAN network

Fig. S4 shows the three noise-aware training approaches we proposed in the main text, the IC-GAN, the WC-GAN and the CR-GAN respectively. All three approaches are based on the off-line training configuration that we first train the GAN on a digital computer. After the network is trained, we mapped the obtained parameters to the photonic hardware for further implementation.

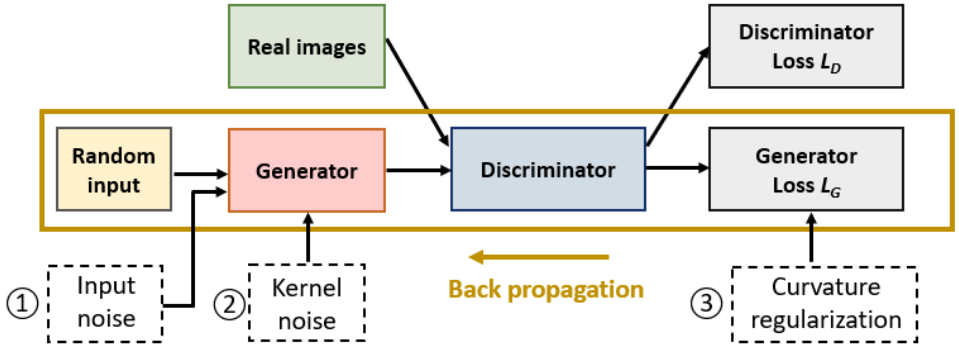

Fig. S4 Schematic of the noise-aware training approaches for the generator in GAN. The parameters in the discriminator are held constant while the parameters in the generator are updated through the backward-propagation process. In the forward propagation process, the noise is added on the (1) random input (for IC-GAN) or (2) the kernel weights (for WC-GAN) to enhance the stability of the GAN against the practical noise. An alternative approach is to (3) introduce the curvature regularization term in the loss function (CR-GAN) to avoid dropping into a local minimum that is sensitive to weight variation.

The IC-GAN approach inflates the STD of the random signal input, 0.5 in our case, during training of the network, while all the other steps in the forward and backward-propagation pass such as the loss function calculation, the gradient descent and weight update, are the same as the conventional GAN training algorithm. We control the learning rates for the discriminator and the generator are the same value,1×10^{-4}, to avoid one overpowering the other. Once all the parameters are obtained, we implement the network in practice only use noise with a smaller STD (0.2 in our experiment) as the input.

For the WC-GAN approach, we cast the effect of all the error sources into the single 5% STD Gaussian noise ΔW^l_{ij} and add the noise onto the corresponding kernel weight at each forward-propagation pass. As shown in Fig. S5a, the introduced noise on kernel weight gives a deviation of the loss function L'_G from its ideal value L_G thus leads to a different gradient. The WC-GAN

performs noiseless gradient descent and weight update in the back-propagation pass based on the L'_G. According to the mapping condition, $\Delta W_{ij}^{l} = \frac{|W|_{max}^{l}}{|\Gamma|_{max}^{l}} \cdot \Delta\Gamma_{ij}^{l}$, the total noise injected ΔW^{l}_{ij} on corresponding kernel weight during training is closely related to the maximum weight absolute value, thus the noise may grow uncontrollably with growing maximum weight values and prevent the training to converge. Therefore we clip the kernel weight distribution in the desired range by rescaling the kernel element with the maximum absolute value of each layer by a factor (of 0.995 in the training) after every weight update. The weight clipping process improves the GAN training convergence. All the other steps in the WC-GAN training approach are the same as the conventional algorithm.

For the CR-GAN approach (see Fig. S4b), we assume the robustness to practical noise of the model can be maximized if the distance between the weight gradient at the point $\mathbf{W}^{l}$ in parametric hyperspace and the gradient of neighborhood points $\mathbf{W}^{l'}$ are minimized. We define the regularization term L_r as the maximal distance between the weight gradient and the gradient of neighborhood point $L_r = \left\| \frac{dL_G}{d\mathbf{W}^{l}} - \frac{dL'_G}{d\mathbf{W}^{l'}} \right\|$ where the range of neighborhood is defined by the discretization step h. In each training step, the regularization term is added to the total loss $L_{wr}=L_G+rL_r$ where r is the regularization strength and then the back-propagation is performed on the weights. In our simulation, the h and r vary under various noise levels to obtain optimal performance.

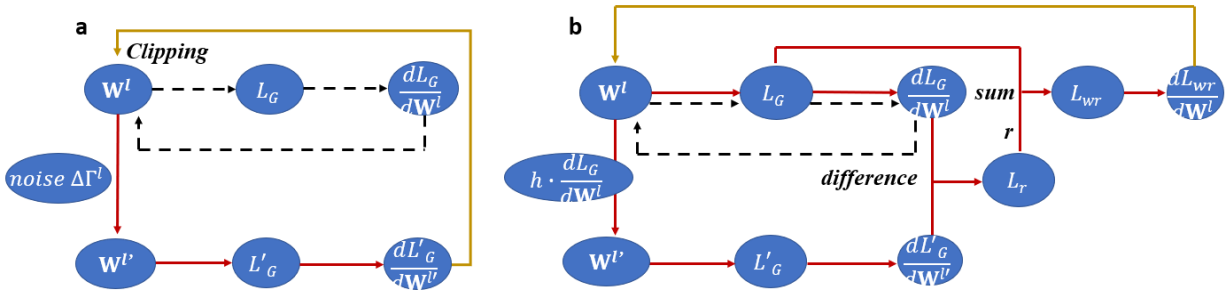

Fig. S5 Schematic of the kernel weight update process for **(a)** WC-GAN and **(b)** CR-GAN respectively. The solid red line and the solid yellow line indicate the forward-propagation pass and the backward-propagation pass respectively. The dashed black line indicates the conventional GAN training process.

V. The architecture of the GAN to generate the handwritten number

We use two GAN networks to generate the handwritten number "7" and the full 10 digits from"0" to "9" respectively.

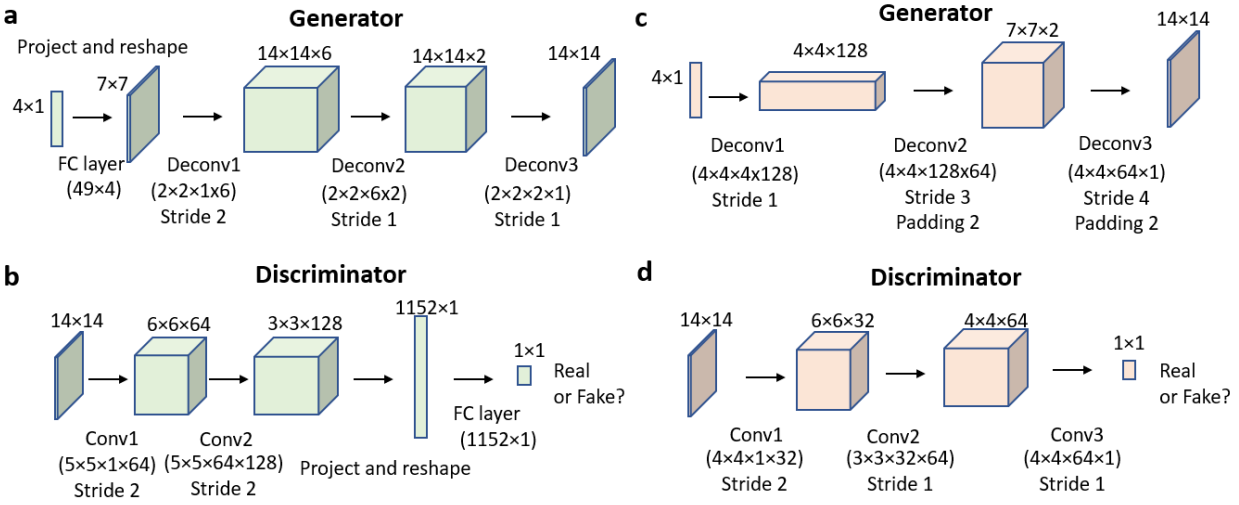

Fig. S6 a and **b**: the architecture of the (a) generator and (b) discriminator in original GAN, WC-GAN and IC-GAN that are used to generate handwritten number "7". **c** and **d**: the architecture of the (c) generator and (d) discriminator in original GAN and CR-GAN that are used to generate the full handwritten digits from "0" to "9".

The generator models in the GAN (see Fig. S6a and S6c) are composed of the fully connected layer (FC) and deconvolution layers (Deconv). After each hidden layer, batch normalization is operated before applying the nonlinear function. For the generator to generate "7" (Fig. S6a), we choose the LeakyReLu as the nonlinear function after each hidden layer and for the generator to generate full digits (Fig. S6c), we choose the ReLu as the nonlinear function. For both models, we use the hyperbolic tangent function as the nonlinear function at the final output.

The discriminators in the GAN (see Fig. S6b and S6d) are composed of the FC layer (FC) and convolution layers (Conv). For both discriminator models, we choose the LeakyReLu as the nonlinear function after each hidden layer and apply the sigmoid function at the final output.

VI. The hand-written numbers generated using accurate kernel and inaccurate kernel matrices.

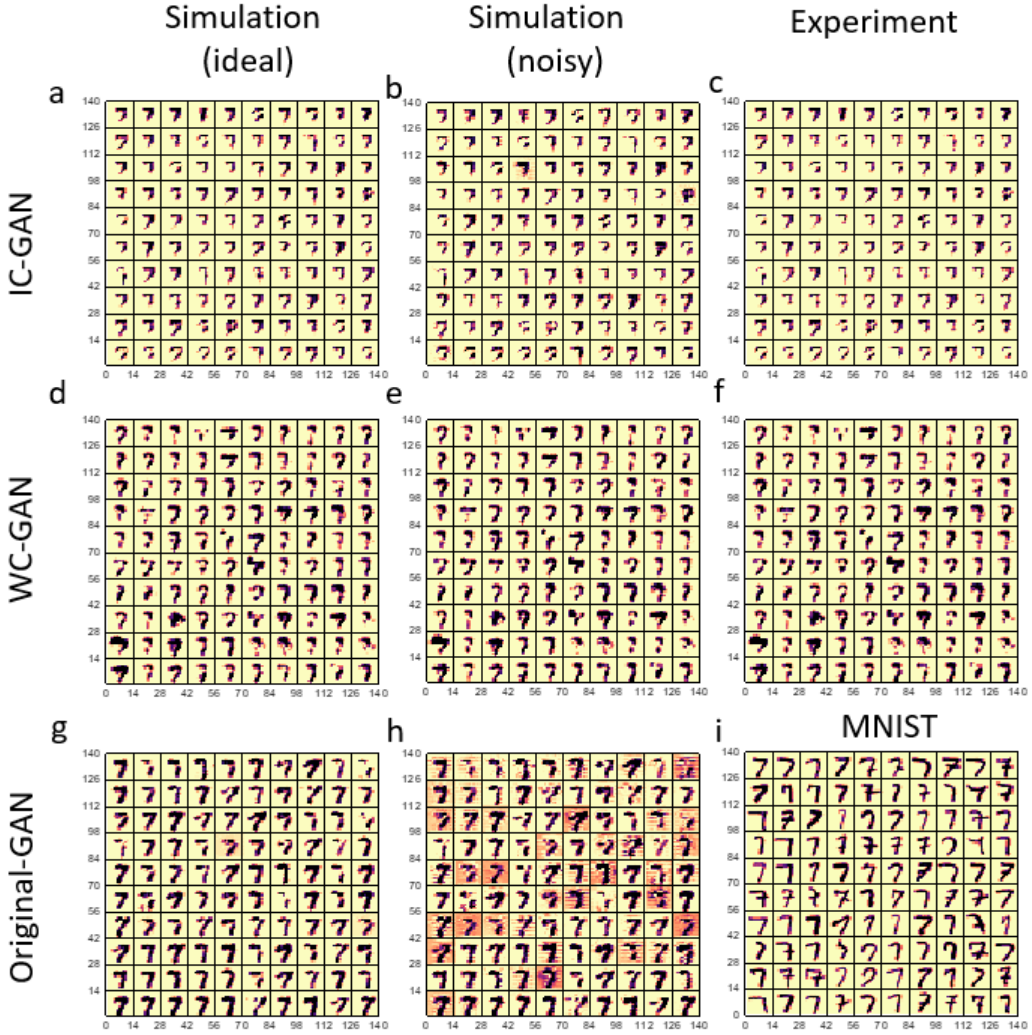

Fig. S7 Images of the handwritten number "7" generated by **(a-c)** the IC-GAN, **(d-f)** the WC-GAN and **(g-h)** the original GAN using the **(i)** MNIST as the training database. The results show that the IC-GAN and WC-GAN are less vulnerable to practical noise.

Fig. S7 shows the images of the handwritten number "7" generated by the IC-GAN (see Fig. S7a-S7c), the WC-GAN (see Fig. S7d-S7f) and the original GAN (Fig S7g, S7h) respectively using the (i) MNIST dataset as the training databased. The simulation results assuming the noisy kernel weights (Fig. S7b and S7e) are consistent with our experimental results (Fig. S7c and S7f). The results also show that compared to the original GAN, the IC-GAN and WC-GAN are less vulnerable to practical noise. The images of the full 10 digits from "0" to "9" generated by the

original GAN and the CR-GAN under the mode contrast setting error ranging from 0% to 10% are shown in Fig. S8 and Fig. S9.

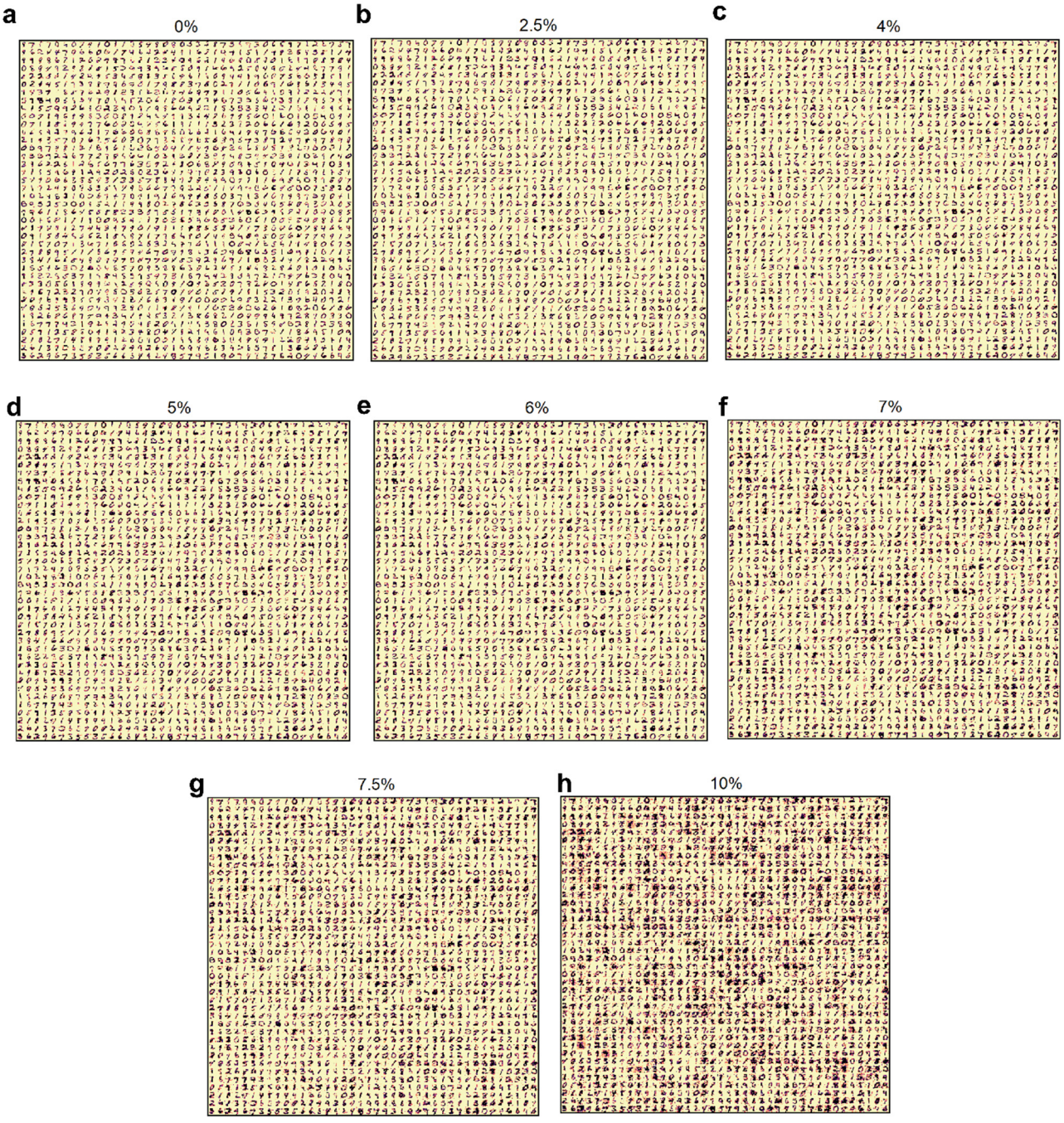

Fig. S8 The images of the full 10 digits from "0" to "9" generated by the CR-GAN in simulation assuming the mode contrast setting error is (a) 0% (ideal), (b) 2.5%, (c) 4%, (d) 5%, (e) 6%, (f) 7% (g) 7.5% and (h) 10%, respectively.

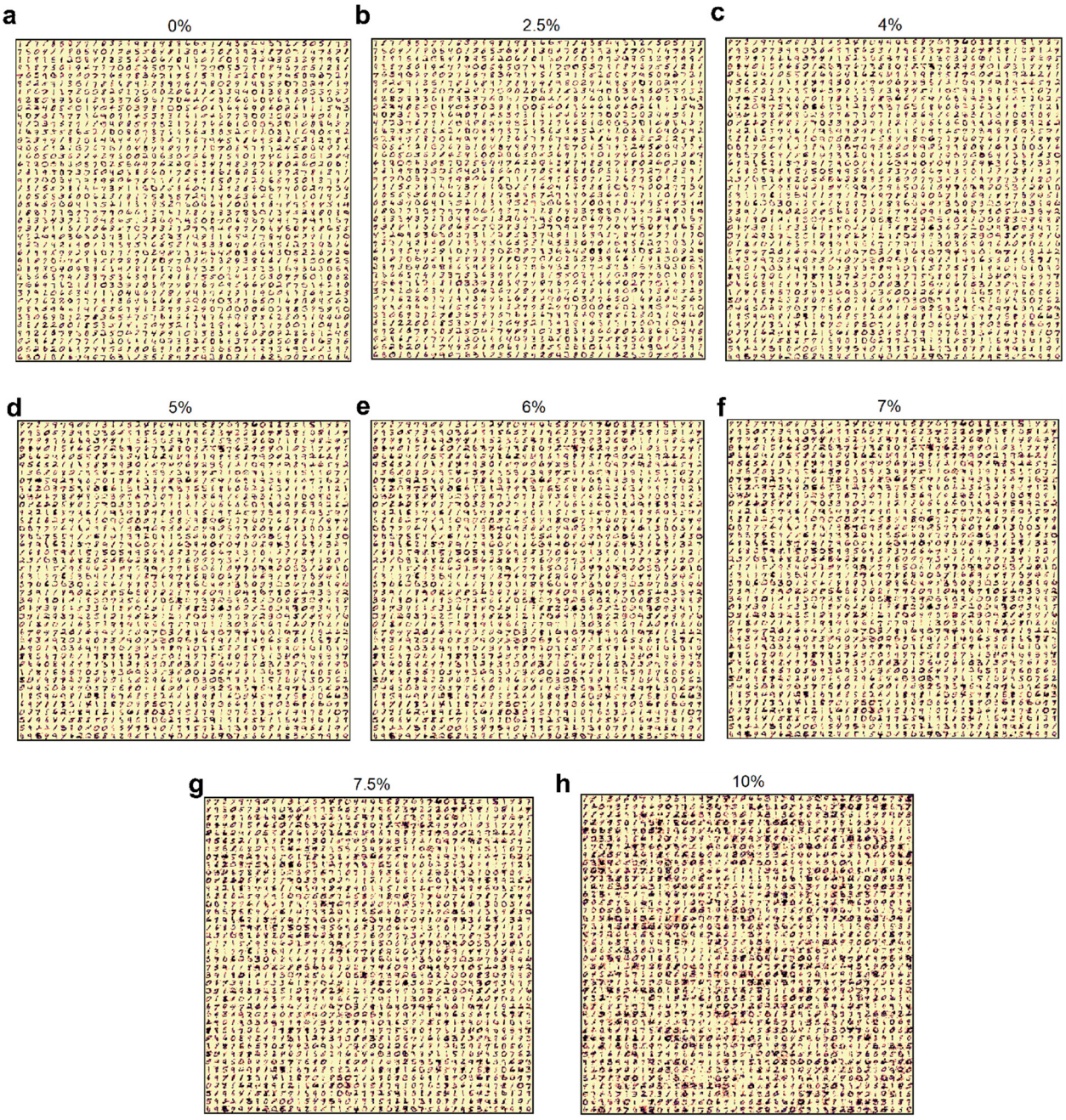

Fig. S9 The images of the full 10 digits from “0” to “9” generated by the CR-GAN in simulation assuming the mode contrast setting error is (a) 0% (ideal), (b) 2.5%, (c) 4%, (d) 5%, (e) 6%, (f) 7% (g) 7.5% and (h) 10%, respectively.

VII. Calculate FID and estimate the diversity of the generated images

The Frechet inception distance (FID) is a metric for evaluating the quality of generated images from both fidelity and diversity and is proposed to specifically evaluate the performance of generative adversarial networks by Martin Heusel et al. in 2017[3]. To calculate FID, we design another CNN as shown in Fig. S10 to classify the generated handwritten digits. The overall error rate of this CNN is only 2.43% after training. When a 14×14 image is sent into this network, the activation features obtained before the last fully connected layer is reshaped as the "feature vector" with the size 160 ×1 and the n feature vectors obtained after n images fed into the CNN are further combining together to form a feature matrix with the size of 160 × n. The FID evaluates generated images by statistically comparing the generated images with the real images from the target domain. Assuming the matrice **X** and **Y** are the feature matrices for the GAN generated images and the real images from MNIST database respectively, the FID then is defined as:

$$FID = \|\mu_X - \mu_Y\|^2 + tr\left(\Sigma_X + \Sigma_Y - 2(\Sigma_X\Sigma_Y)^{\frac{1}{2}}\right),$$

where μ_X and μ_Y refer to the feature-wise mean vectors of the GAN-generated images and real images, Σ_X and Σ_Y are the covariance matrices of the corresponding feature matrice **X** and **Y,** and "*tr*" refers to the trace operation. To perform more accurate FID results in this work, we also break the generated images into groups and calculate the mean and standard deviation of the FID.

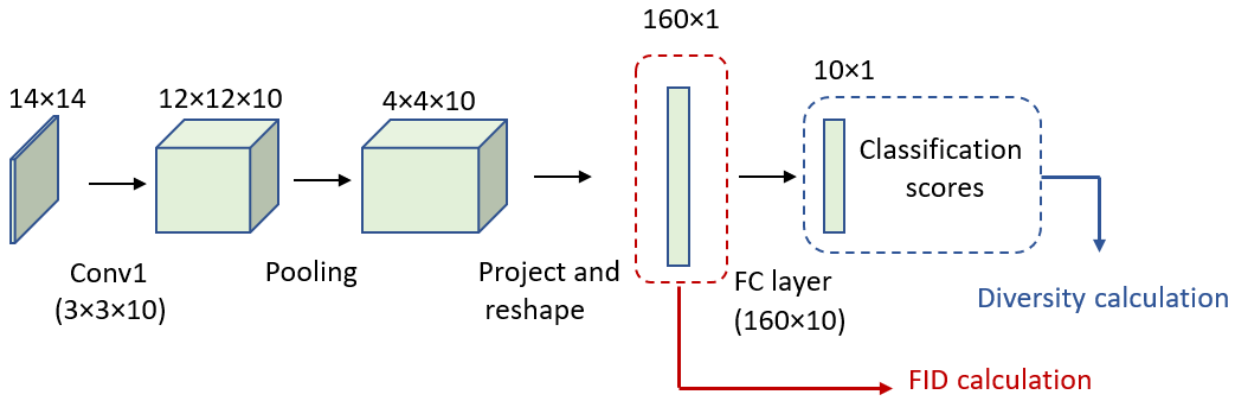

Fig. S10 The architecture of the convolutional neural network is used for handwritten digits classification. When a 14×14 image is sent into this network, the activation features obtained before the last fully connected layer is reshaped as the "feature vector" with the size 160 ×1 to calculate the FID. The statistics of the classification results are used to estimate the diversity.

We also estimate the diversity of generated images using the diversity coefficient, which is defined as the STD of the percentage of each number classes in the generated images:

$$Diversity\ Coefficient = \sqrt{\frac{\sum_{i=0}^{9}\left|P_i - \frac{1}{10}\right|^2}{10}}.$$

Where P_i is the statistic percentage for the generated image is successfully classified as the number "i", i is the digit number from 0 to 9. The more diversity the generated images have, the more uniform the percent of ten classes and the lower the diversity coefficient is. For example, for the MNIST training database which has a close-to-uniform percent of ten classes, the diversity coefficient is as low as 0.00585 (see the solid black horizontal line in Fig. S11). Fig. S11 shows the diversity coefficient of images generated by the original GAN and the CR-GAN as a function of the mode contrast error. As the noise level increases from 0% (ideal) to 8%, the diversity coefficient gradually drops, which supports our claim in the main text that the diversity of the generated images increases with a larger noise. The increase of the diversity coefficient at 10% mode contrast setting error is due to the breakdown of the classification accuracy of the CNN, the quality of the generated images is so bad that CNN can not give the correct class of the generated image.

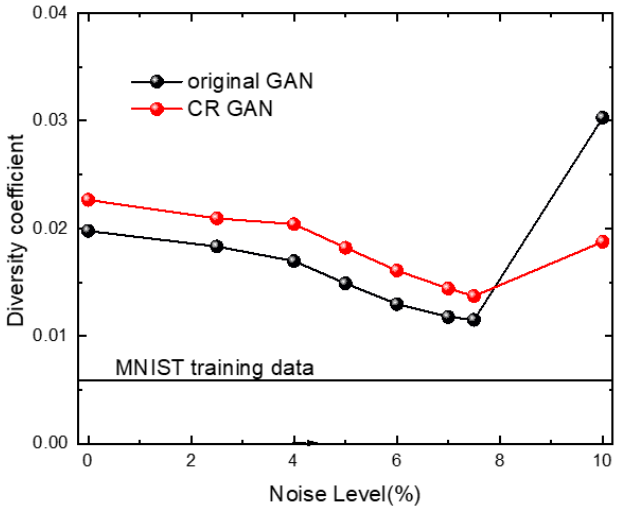

Fig. S11 The diversity coefficient of the generated images by original GAN and CR-GAN respectively under various effective noise $\Delta\Gamma^{I}_{ij}$ with STD ranging from 0% to 10%.

Reference:

1. Williams, C. R. S., Salevan, J. C., Li, X., Roy, R. & Murphy, T. E. Fast physical random number generator using amplified spontaneous emission. *Optics Express* **18**, 23584–23597 (2010).

2. Wu, C. *et al.* Programmable phase-change metasurfaces on waveguides for multimode photonic convolutional neural network. *Nature Communications* **12**, 96 (2021).

3. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *arXiv preprint arXiv:1706.08500* (2017).

4. Yudeng Lin *et al.* Demonstration of Generative Adversarial Network by Intrinsic Random Noises of Analog RRAM Device. in *IEEE International Electron Devices Meeting (IEDM)* 3–4 (2018).