Research Article

ADVANCED PHOTONICS

Diffraction casting

Ryosuke Mashiko, Makoto Naruse,† and Ryoichi Horisaki*
The University of Tokyo, Graduate School of Information Science and Technology, Department of Information Physics and Computing, Tokyo, Japan

Abstract. Optical computing is considered a promising solution for the growing demand for parallel computing in various cutting-edge fields that require high integration and high-speed computational capacity. We propose an optical computation architecture called diffraction casting (DC) for flexible and scalable parallel logic operations. In DC, a diffractive neural network is designed for single instruction, multiple data (SIMD) operations. This approach allows for the alteration of logic operations simply by changing the illumination patterns. Furthermore, it eliminates the need for encoding and decoding of the input and output, respectively, by introducing a buffer around the input area, facilitating end-to-end all-optical computing. We numerically demonstrate DC by performing all 16 logic operations on two arbitrary 256-bit parallel binary inputs. Additionally, we showcase several distinctive attributes inherent in DC, such as the benefit of cohesively designing the diffractive elements for SIMD logic operations that assure high scalability and high integration capability. Our study offers a design architecture for optical computers and paves the way for a next-generation optical computing paradigm.

Keywords: optical computing; diffractive neural network; SIMD operations; parallel computing; logic operations; machine learning.

Received May 12, 2024; revised manuscript received Aug. 16, 2024; accepted for publication Sep. 3, 2024; published online Oct. 3, 2024.

[DOI: 10.1117/1.AP.6.5.056005]

1 Introduction

Optical computing is a longstanding and captivating topic in the fields of optics and photonics. It is considered a potential post-Moore computing technology[1] that offers distinct advantages, including high bandwidth, rapid processing speed, low power consumption, and parallelism.[2,3] Around the 1980s, optical computing was actively explored, with developments in technologies, such as optical vector matrix multipliers[4–7] and optical associative memories.[8–10] Among these, shadow casting (SC) emerged as a prominent optical computing technology of that era.[11–14] SC facilitated single instruction, multiple data (SIMD) for logical operations through optical and spatially parallel computing. The SC scheme relied on shadowgrams, which optically generated a single output image through massively parallel logic operations from two binary input images. The versatility of SIMD logic operations was attained by altering the illumination pattern of the shadowgrams. Another key aspect involved the computational encoding and decoding of input and output images, respectively, designed to balance light intensities between the zeros and ones in the binary images. This computational process was an obstacle in achieving end-to-end optical computing. Despite the anticipated benefits in speed and energy efficiency, these optical computing technologies in the 1980s stagnated due to limitations in hardware (fabrication) and software (design) for optical components at that time. As a result, they lagged behind the major progress made in electronic computing.

Over the past few decades, significant advancements in microfabrication, mathematical optimization, and computational power have dramatically transformed the field of optical computing from what it was in the 1980s. Several pioneering optical computing techniques have been studied, including waveguide-based photonic circuits and diffractive neural networks (DNNs). Waveguide-based photonic circuits, which integrate waveguide interferometers, have high compatibility with currently existing electronic computers and circuits. They have led to a wide range of applications, such as vector-matrix operations,[15–17] logical operations,[18–24] and integrated reconfigurable circuits.[25–27]

DNNs consist of cascaded diffractive optical elements (DOEs), which emulate neural network connections as light waves pass through the DOEs. This configuration utilizes the spatial parallelism of light and realizes fast and energy-efficient computation. A wide range of attractive applications based on DNNs has been

*Address all correspondence to Ryoichi Horisaki, horisaki@g.ecc.u-tokyo.ac.jp
†Deceased.

proposed, including image classification,[28–33] image processing,[34–37] linear transformations,[38–40] and logic operations.[41–47]

Currently, the demand for computation in SIMD logic operations has intensified, particularly due to advancements in cutting-edge technologies such as image processing, machine learning, and blockchain.[48–50] Traditional computation with central processing units is often inadequate to meet the computational needs of these fields. Consequently, graphics processing units,[51] tensor processing units,[52] field-programmable gate arrays (FPGAs),[53,54] and application-specific integrated circuits[55,56] are employed as SIMD-specific devices. The trend towards high-speed, energy-efficient, and massively parallel computing aligns well with the advantages of optical computation. As a result, there has been a rapid increase in efforts to develop practical optical computing methods for SIMD logic operations. Optical SIMD logic operations have been achieved using waveguide photonic circuits.[26,27] However, a drawback of this approach is its limited scalability, which arises from the need for precise yet large-scale fabrication.

The use of DNNs holds potential as a solution to this issue, owing to the parallelism inherent in free-space propagation. For instance, several types of DNN-based logic operations have been proposed to overcome this drawback,[41–47] but these previous methods typically involve only one or a few types of logic operations on a small number of bits, and the realization of SIMD logic operations using DNNs remains unachieved. Moreover, these methods still require computational encoding and decoding of the input and output, respectively, posing a significant challenge for end-to-end optical computing, similar to the SC scheme.

In this study, considering the background mentioned above, we present a method termed diffraction casting (DC) for conducting all 16 optical SIMD logic operations on more than one hundred bits by incorporating the SC scheme and DNNs. DC revives SC through the use of DNNs. Therefore, DC shares the motivation of SC but exhibits several differences and advantages over SC. Unlike SC, which is based on geometrical optics, DC is grounded in wave optics. As a result, DC incorporates wave phenomena such as diffraction and interference through the use of DOE cascades in DNNs and is anticipated to offer greater integration capability compared to SC. Another advantage of DC is its elimination of the need for computational encoding and decoding of the input and output, respectively, which have been inherent bottlenecks in the SC scheme and previous DNN-based logic operations referred to above. This is enabled by introducing a buffer area around the input pair. In the rest of the paper, we will elaborate on the architectural design of DC, including the forward model, the optimization process, and provide numerical demonstrations.

2 Materials and Methods

2.1 Concept of Diffraction Casting

DC is designed for 16 types of SIMD logic operations, processing two input binary images to produce one output binary image. Figure 1 depicts the conceptual architecture of DC. DC consists of a reconfigurable illumination, DOEs, and an input layer. The reconfigurable illumination enables switching between logical operations and casts light on the DOE cascade forming a DNN. In this paper, we focus on the reconfigurable illumination with binary amplitude modulation implemented using a digital micromirror device (DMD) illuminated with coherent light from a laser, and on DOEs with phase modulation, assuming the use of commercially available optical components.

We place the two input images side by side on the input layer within the DOE cascade to achieve a simple optical setup. The output of the logic operation appears as an intensity distribution at the end of the cascade and is captured with an image sensor. The final result is binarized by assuming a one-bit image sensor or a computational process. The reconfigurable illumination and DOEs are specifically trained to perform the 16 SIMD logic operations on any two binary images, as detailed in the subsequent subsection. Once the training process is completed, DC enables massively parallel optical logic operations on arbitrary binary inputs just by selecting the illumination patterns, without necessitating any modifications to the DOEs.

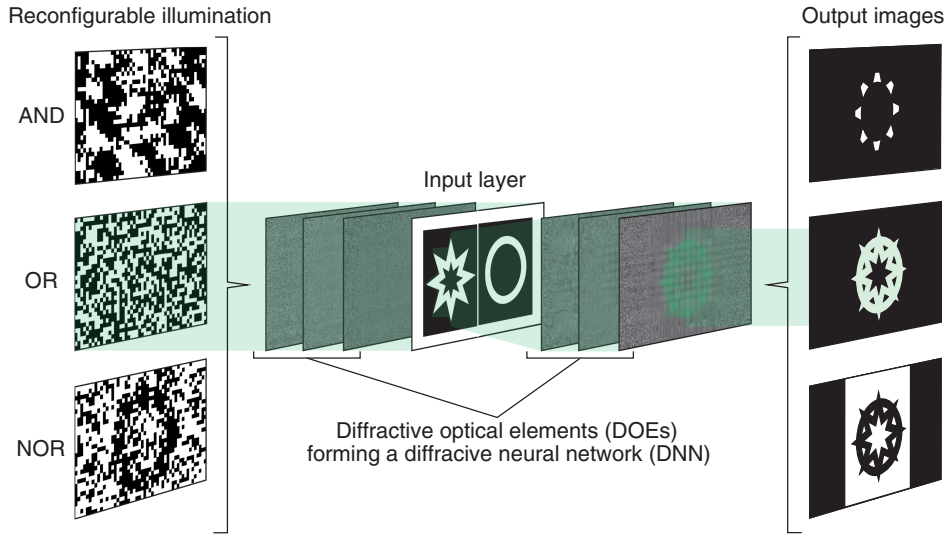

Fig. 1 Schematic diagram of DC. The selection of a logic operation is performed using reconfigurable illumination without any modification to the DOEs.

2.2 Optical Forward Model

Figure 2 illustrates the forward and backward processes of DC. We consider a total of L types of SIMD logic operations on N parallel bits. These operations are conducted by an optical cascade composed of K layers, including one illumination layer, one input layer, and $K-2$ DOE layers, with the layer index denoted as $k \in \{1, 2, \ldots, K\}$. The first layer of the optical cascade is the binary reconfigurable illumination $\boldsymbol{r}_l \in \{0,1\}^{P_x \times P_y}$, where $l \in \{1, 2, \cdots, L\}$ is the index of the logic operations. An input pair $\boldsymbol{f} \in \{0,1\}^{N_x \times 2N_y}$, composed of side-by-side binary images, is located on the input layer, denoted as the K_{in}'th layer in the cascade.

Here, N_x and N_y represent the pixel counts of the individual images within the input pair along the x and y directions, respectively, where $N_x \times N_y = N$. The phase distributions for each DOE layer are denoted by $\boldsymbol{\phi}_k \in \mathbb{R}^{P_x \times P_y}$, and P_x and P_y indicate the pixel counts of the DOEs along the x and y axes, respectively. The result of the logic operations is observed with the image sensor located downstream of the K'th layer in the optical cascade.

We now describe the forward process of DC. The complex amplitude modulation $\boldsymbol{v}_k \in \mathbb{C}^{P_x \times P_y}$, induced by the reconfigurable illumination, input pair, and the phase-only DOEs at the k'th layer in the cascade, is expressed as

$$\boldsymbol{v}_k = \begin{cases} \boldsymbol{r}_l & \text{for } k = 1, \\ \mathcal{I}[\boldsymbol{f}] + \boldsymbol{t} & \text{for } k = K_{\text{in}}, \\ \exp(j\boldsymbol{\phi}_k) & \text{otherwise}, \end{cases} \tag{1}$$

where j denotes the imaginary unit. Here, $\mathcal{I}$ is an operator transforming the input pair into an amplitude image on the input layer, composed of the following two steps. The first step is upsampling along the x and y directions with factors of $s_x \in \mathbb{N}$ and $s_y \in \mathbb{N}$, respectively. The second step is zero padding to enlarge the upsampled input pair to the DOE size ($P_x \times P_y$ pixels).

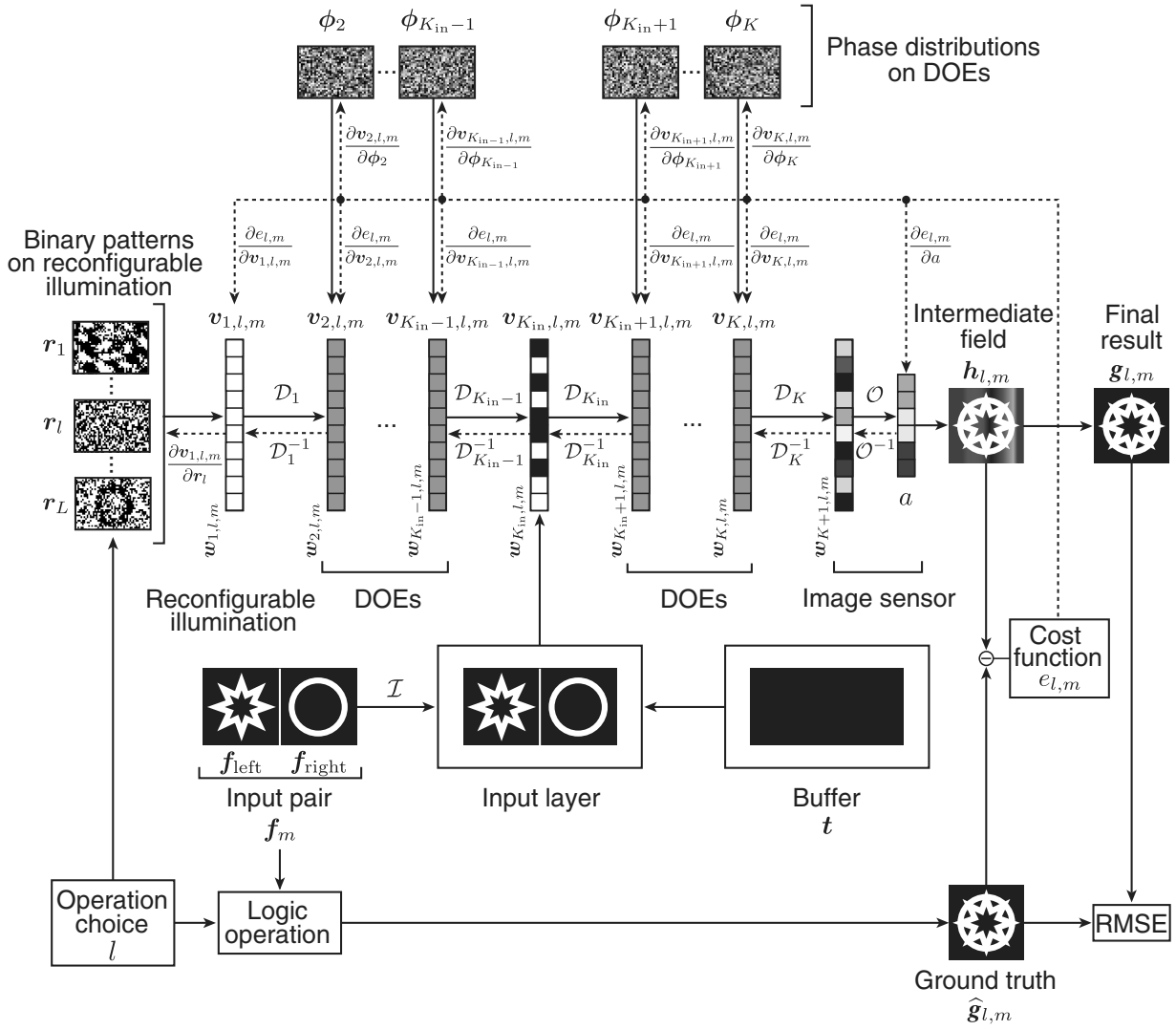

Fig. 2 Forward and backward processes of DC. The reconfigurable illumination, the DOEs, and the scaling factor are optimized through the training process.

$\boldsymbol{t} \in \{0,1\}^{P_x \times P_y}$ expresses a buffer surrounding the upsampled input pair, as illustrated in Fig. 2 and is defined as follows:

$$\boldsymbol{t}(u_x, u_y) = \begin{cases} 0 & \text{for } \{u_x, u_y\} \in \left(\frac{P_x - s_x N_x}{2}, \frac{P_x + s_x N_x}{2}\right] \times \left(\frac{P_y - 2s_y N_y}{2}, \frac{P_y + 2s_y N_y}{2}\right], \\ 1 & \text{otherwise,} \end{cases} \tag{2}$$

where $u_x \in \mathbb{N}$ and $u_y \in \mathbb{N}$ are indices along the x and y directions. This buffer is employed to compensate for light intensities transmitted or blocked on the input pair and enables the removal of computational encoding and decoding of the input and output, processes that are indispensably employed in previous optical logic operation methods, including the SC scheme.

The propagation process passing through the k'th layer in the cascade is written as

$$\boldsymbol{w}_{k+1} = \mathcal{D}_k[\boldsymbol{v}_k \boldsymbol{w}_k], \tag{3}$$

where $\boldsymbol{w}_k \in \mathbb{C}^{P_x \times P_y}$ is a complex amplitude field just before the k'th layer. $\mathcal{D}_k$ is a diffraction operator representing the propagation from the k'th layer to the $(k+1)$'th layer, calculated based on the angular spectrum method.[57] The initial field $\boldsymbol{w}_1$ is specified as an all-ones matrix, indicating a uniform field at the start.

The output intensity field of the optical cascade is observed with the image sensor as follows:

$$\boldsymbol{h} = a\mathcal{O}[|\boldsymbol{w}_{K+1}|^2], \tag{4}$$

$$\boldsymbol{g} = \mathcal{B}[\boldsymbol{h}]. \tag{5}$$

Here, $\boldsymbol{h} \in \mathbb{R}^{P_x \times P_y}$ represents the intermediate field, obtained through an operator $\mathcal{O}$ that first crops the central $s_x N_x \times s_y N_y$ pixels from the output intensity field and then downsamples it to the original input image size of $N_x \times N_y$. This process includes scaling the intensity by a factor of $a \in \mathbb{R}_{>0}$, which corresponds to either amplifying or attenuating the signal. $\boldsymbol{g} \in \mathbb{R}^{N_x \times N_y}$ is the final result of the logic operation, with the binarization operator $\mathcal{B}$ defined as

$$\mathcal{B}[b] = \begin{cases} 0 & \text{for } b < 0.5, \\ 1 & \text{for } b \geq 0.5, \end{cases} \tag{6}$$

where $b \in \mathbb{R}$ is an arbitrary variable. The binarization process, which converts analog signals to Boolean ones, is implemented using either a one-bit image sensor or through computational means. The variation of the threshold in the binarization is not crucial to the performance of DC. This is because the scaling factor a in Eq. (4) is optimized for the threshold through the gradient descent process, as described in the next section, and thus neutralizes the impact of the threshold.

2.3 Optimization Process

To realize optical logic operations in parallel, the illumination $\boldsymbol{r}_l$, the DOEs $\boldsymbol{\phi}_k$, and the scaling factor a are optimized based on gradient descent in this study. First, we describe the optimization process by assuming a single logic operation ($L = 1$) and a single input pair for simplicity, where the illumination is defined as $\boldsymbol{r}_{L=1}$. Then, we extend the optimization process to arbitrary numbers of logic operations and input pairs.

2.3.1 Derivatives for a single logic operation and a single input pair

We define a cost function e for a single logic operation and a single input pair based on the mean squared error (MSE) as follows:

$$e = \frac{1}{N} \sum_{\forall} |\boldsymbol{e}|^2, \tag{7}$$

where $\sum_{\forall}$ represents the summation of all the elements of a tensor on its right side. Here, the error $\boldsymbol{e}$ is defined by

$$\boldsymbol{e} = \boldsymbol{h} - \hat{\boldsymbol{g}}, \tag{8}$$

which represents the difference between the intermediate field $\boldsymbol{h}$ and the ground truth of the operation result $\hat{\boldsymbol{g}}$. This is to avoid the intermediate field's signals around the threshold values in $\mathcal{B}$, ensuring a robust binarization process.

To optimize $\boldsymbol{r}_{L=1}$ and $\boldsymbol{\phi}_k$ based on gradient descent, the partial derivatives of e with respect to these variables are expressed by employing the chain rule as follows:

$$\frac{\partial e}{\partial \boldsymbol{r}_{L=1}} = \frac{\partial \boldsymbol{v}_1}{\partial \boldsymbol{r}_{L=1}} \cdot \frac{\partial e}{\partial \boldsymbol{v}_1}, \tag{9}$$

$$\frac{\partial e}{\partial \boldsymbol{\phi}_k} = \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{\phi}_k} \cdot \frac{\partial e}{\partial \boldsymbol{v}_k}. \tag{10}$$

The right sides of these partial derivatives include the partial derivative of e with respect to $\boldsymbol{v}_k$, calculated as

$$\frac{\partial e}{\partial \boldsymbol{v}_k} = \frac{4a}{N} \boldsymbol{w}_k^* \mathcal{D}_k^{-1}[\boldsymbol{v}_{k+1}^* \mathcal{D}_{k+1}^{-1}[\cdots[\boldsymbol{v}_K^* \mathcal{D}_K^{-1}[\boldsymbol{w}_{K+1} \mathcal{O}^{-1}[\boldsymbol{e}]]]\cdots]], \tag{11}$$

where $\mathcal{D}_k^{-1}$ and $\mathcal{O}^{-1}$ are operators representing the inverse processes of $\mathcal{D}_k$ and $\mathcal{O}$, respectively, and the superscript * denotes the complex conjugate.

The partial derivatives with respect to each optimized variable are finally written as follows. The partial derivative with respect to $\boldsymbol{r}_{L=1}$ is described as

$$\frac{\partial e}{\partial \boldsymbol{r}_{L=1}} = \mathrm{Re}\left[\frac{\partial e}{\partial \boldsymbol{v}_1}\right], \tag{12}$$

where $\mathrm{Re}[\cdot]$ denotes the real part of a complex amplitude. The partial derivative with respect to $\boldsymbol{\phi}_k$ is described as

$$\frac{\partial e}{\partial \boldsymbol{\phi}_k} = \mathrm{Re}\left[-j\boldsymbol{v}_k^* \frac{\partial e}{\partial \boldsymbol{v}_k}\right]. \tag{13}$$

The partial derivative with respect to a is described as

$$\frac{\partial e}{\partial a} = \frac{2}{aN} \sum_{\forall} \boldsymbol{h}\boldsymbol{e}. \tag{14}$$

2.3.2 Derivatives for multiple logic operations and multiple input pairs

Next, we extend the optimization process from a single logic operation and a single input pair as described above, to L logic operations and M input pairs. In this scenario, the cost function E based on the MSE is expressed as

$$E = \frac{1}{LM}\sum_{l,m} e_{l,m}, \tag{15}$$

where $e_{l,m}$ denotes the cost associated with the l'th logic operation and the m'th input pair, derived from Eq. (7). The partial derivatives of E with respect to $\boldsymbol{r}_l$, $\boldsymbol{\phi}_k$, and a are presented as summations of the partial derivatives of $e_{l,m}$ with respect to these variables, derived from Eqs. (12)–(14), respectively,

$$\frac{\partial E}{\partial \boldsymbol{r}_l} = \sum_{m} \frac{\partial e_{l,m}}{\partial \boldsymbol{r}_l}, \tag{16}$$

$$\frac{\partial E}{\partial \boldsymbol{\phi}_k} = \sum_{l,m} \frac{\partial e_{l,m}}{\partial \boldsymbol{\phi}_k}, \tag{17}$$

$$\frac{\partial E}{\partial a} = \sum_{l,m} \frac{\partial e_{l,m}}{\partial a}. \tag{18}$$

2.3.3 Updating procedure

The variables $\boldsymbol{r}_l$, $\boldsymbol{\phi}_k$, and a are updated with the partial derivatives in Eqs. (16)–(18) based on the Adam optimizer.[58] The updating processes for the DOEs $\boldsymbol{\phi}_k$ and the scaling factor a are described as follows:

$$\boldsymbol{\phi}_k \leftarrow \boldsymbol{\phi}_k - \mathrm{Adam}\left[\frac{\partial E}{\partial \boldsymbol{\phi}_k}\right], \tag{19}$$

$$a \leftarrow a - \mathrm{Adam}\left[\frac{\partial E}{\partial a}\right], \tag{20}$$

where Adam$[\cdot]$ represents an operator to calculate the updating step in the Adam optimizer with the derivatives. To simplify the physical realization of the illumination $\boldsymbol{r}_l$, we assume its binary implementations, such as DMDs, by introducing stochastic perturbations into the gradient descent process.[59] The first step in the update process for the variables is as follows:

$$\tilde{\boldsymbol{r}_l} \leftarrow C\left[\tilde{\boldsymbol{r}_l} - \mathrm{Adam}\left[\frac{\partial E}{\partial \boldsymbol{r}_l}\right]\right], \tag{21}$$

where $\tilde{\boldsymbol{r}}_l$ is an intermediate variable for the backward process in the optimization of $\boldsymbol{r}_l$. Here, $\mathcal{C}$ is an operator for clipping the range of values as follows:

$$\mathcal{C}[b] = \begin{cases} 0 & \text{for } b < 0, \\ 1 & \text{for } b > 1, \\ b & \text{otherwise.} \end{cases} \tag{22}$$

Subsequently, $\boldsymbol{r}_l$ in the forward process is updated as follows:

$$\boldsymbol{r}_l = \mathcal{B}[\tilde{\boldsymbol{r}}_l + \boldsymbol{q}], \tag{23}$$

where $\boldsymbol{q} \in \mathbb{R}^{P_x \times P_y}$ is a uniform distribution between ± 0.5, introduced to avoid local minima in the binary optimization. After the optimization process, $\boldsymbol{r}_l$ is finalized as follows:

$$\boldsymbol{r}_l = \mathcal{B}[\tilde{\boldsymbol{r}}_l]. \tag{24}$$

3 Numerical Demonstration

3.1 Experimental Conditions

We numerically demonstrated DC with all 16 logic operations ($L = 16$) for the Boolean input pair $\boldsymbol{f}_{\text{left}}$ and $\boldsymbol{f}_{\text{right}}$, as shown in Table 1. In this numerical demonstration, the wavelength of the coherent light for the reconfigurable illumination λ was defined as 0.532 μm. The optical cascade comprised eleven layers ($K = 11$), incorporating nine DOEs, with the input layer positioned as the sixth layer ($K_{\text{in}} = 6$). The intervals between the layers were equally set to $3 \times 10^4 \lambda\,(\approx 1.60 \times 10^4\ \mu\text{m})$. For the illumination pattern $\boldsymbol{r}_l$, the DOEs $\boldsymbol{\phi}_k$, and the input layer, the pixel pitch was $16\lambda\,(\approx 8.51\ \mu\text{m})$. This pixel pitch was

Table 1 Logic operations defined on input pair.

Input pair $\boldsymbol{f}_{\text{left}}$ / $\boldsymbol{f}_{\text{right}}$	Operation index l	Logic operation	Boolean 0 0 1 1 / 0 1 0 1
Output $\hat{\boldsymbol{g}}_l$	1	0	0 0 0 0
	2	$\boldsymbol{f}_{\text{left}} \wedge \boldsymbol{f}_{\text{right}}$ (AND)	0 0 0 1
	3	$\boldsymbol{f}_{\text{left}} \wedge \overline{\boldsymbol{f}_{\text{right}}}$	0 0 1 0
	4	$\boldsymbol{f}_{\text{left}}$	0 0 1 1
	5	$\overline{\boldsymbol{f}_{\text{left}}} \wedge \boldsymbol{f}_{\text{right}}$	0 1 0 0
	6	$\boldsymbol{f}_{\text{right}}$	0 1 0 1
	7	$\boldsymbol{f}_{\text{left}} \oplus \boldsymbol{f}_{\text{right}}$ (XOR)	0 1 1 0
	8	$\boldsymbol{f}_{\text{left}} \vee \boldsymbol{f}_{\text{right}}$ (OR)	0 1 1 1
	9	1	1 1 1 1
	10	$\overline{\boldsymbol{f}_{\text{left}} \wedge \boldsymbol{f}_{\text{right}}}$ (NAND)	1 1 1 0
	11	$\overline{\boldsymbol{f}_{\text{left}}} \vee \boldsymbol{f}_{\text{right}}$	1 1 0 1
	12	$\overline{\boldsymbol{f}_{\text{left}}}$	1 1 0 0
	13	$\boldsymbol{f}_{\text{left}} \vee \overline{\boldsymbol{f}_{\text{right}}}$	1 0 1 1
	14	$\overline{\boldsymbol{f}_{\text{right}}}$	1 0 1 0
	15	$\overline{\boldsymbol{f}_{\text{left}} \oplus \boldsymbol{f}_{\text{right}}}$ (XNOR)	1 0 0 1
	16	$\overline{\boldsymbol{f}_{\text{left}} \vee \boldsymbol{f}_{\text{right}}}$ (NOR)	1 0 0 0

chosen by considering commercially available spatial light modulators (SLMs), including DMDs, as well as the microfabrication technique used for configuring DOEs,[60] and the simplicity of using an integer product.[44,45] The pixel count was 160 $(= P_x)$ along the x axis and 288 $(= P_y)$ along the y axis, respectively. For the input pair $\boldsymbol{f}_m$, the pixel count of the individual image in the pair was 16 $(= N_x)$ along the x axis and 16 $(= N_y)$ along the y axis, where the parallel bits N became 256, and the upsampling factors s_x and s_y were both 8. The width of the region with ones on the buffer $\boldsymbol{t}$, as defined in Eq. (2), was set to 16 pixels. To prevent the circulant effect on the diffraction calculation, the complex amplitude fields were zero-padded with a width of 64 pixels during the layer-by-layer propagation processes.

For the optimization process, the input pairs were generated with values initially selected from uniform random distributions between 0 and 1 and were then binarized using randomly selected thresholds, also between 0 and 1. The numbers of input pairs for training and testing were 80,000 and 256, respectively, without any duplication. The batch size M was set to 16 for training. The number of iterations was 5000. The learning rates for the Adam optimizer, as used in Eqs. (19)–(21), for $\boldsymbol{r}_l$, $\boldsymbol{\phi}_k$, and a were set to 3×10^{-2}, 1×10^{-2}, and 3×10^{-3}, respectively. These variables were initially set to uniform random distributions for $\boldsymbol{r}_l$ and $\boldsymbol{\phi}_k$, and 10 for a. The final performance of DC with L logic operations was evaluated by the root mean squared errors (RMSEs) between the final result $\boldsymbol{g}_{l,m}$ and the ground truth $\hat{\boldsymbol{g}}_{l,m}$ for M test input pairs, as shown in Fig. 2 and described as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{LMN}\sum_{l,m}\sum_{\forall}|\boldsymbol{g}_{l,m} - \hat{\boldsymbol{g}}_{l,m}|^2}. \tag{25}$$

3.2 Result

The optimization results for the illumination $\boldsymbol{r}_l$ and the DOEs $\boldsymbol{\phi}_k$ are presented in Figs. 3(a) and 3(b), respectively. The scaling factor a was optimized to 23.8. In the numerical demonstration shown in Fig. 4, DC was performed using two test input pairs with 256 parallel bits. The first pair, shown in Fig. 4(a), was

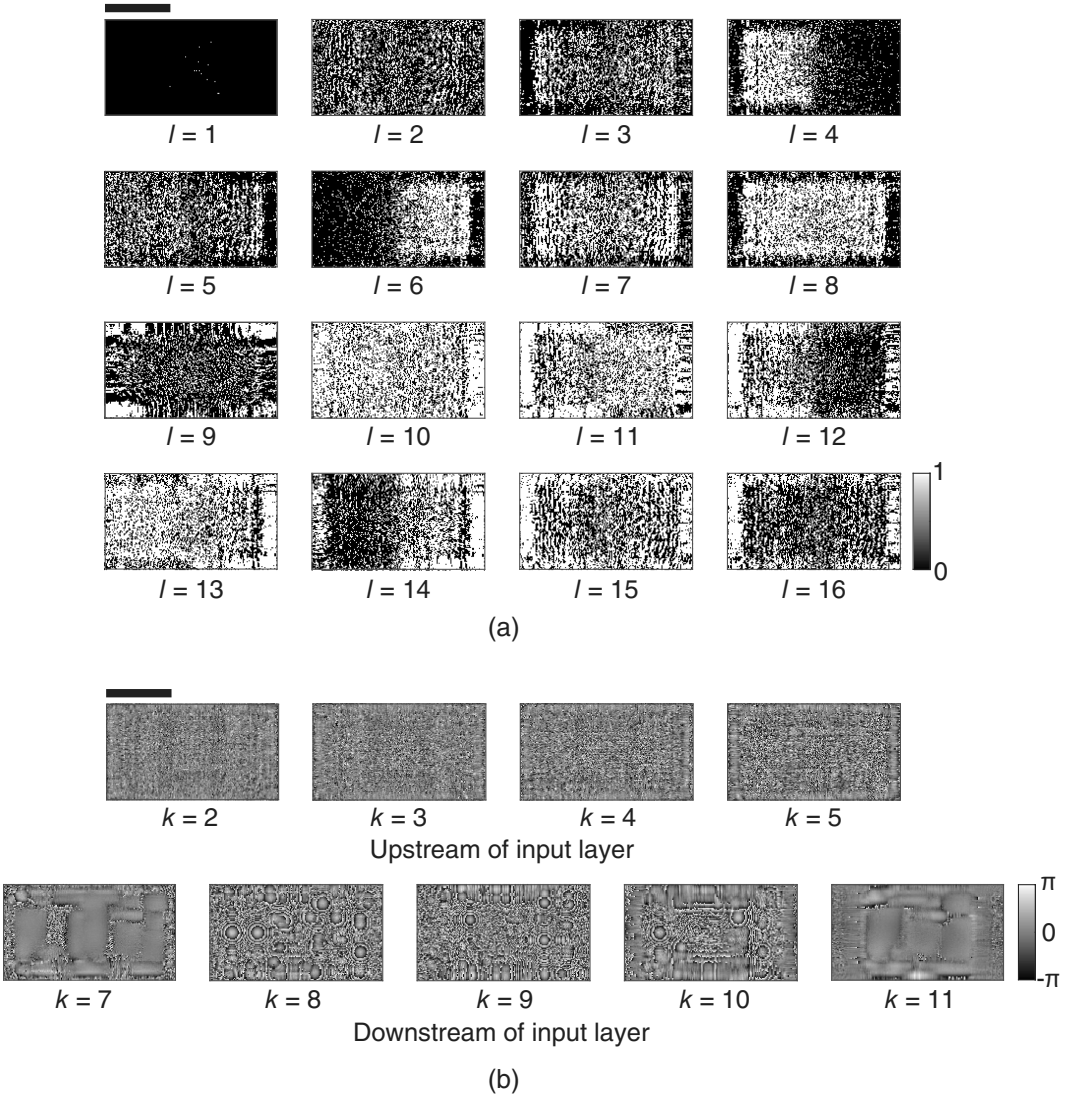

Fig. 3 Optimization results. (a) Binary amplitude patterns on a DMD for the reconfigurable illumination and (b) phase distributions on the DOEs. Scale bar is 1 mm.

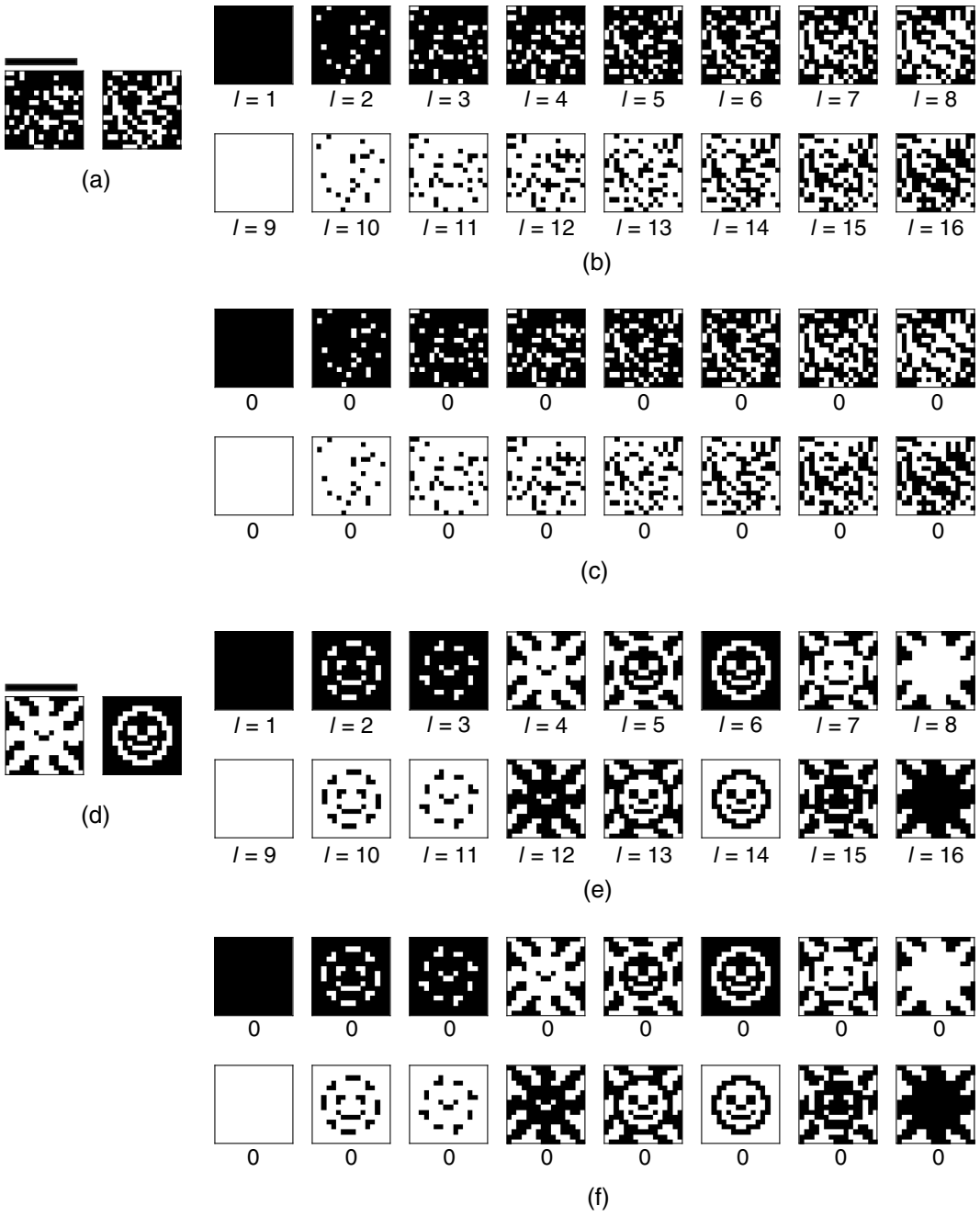

Fig. 4 Examples of the DC process with the optimized illumination and the DOEs shown in Fig. 3. (a) Test input pair of random patterns and its corresponding (b) ground truths of the 16 operations, with the operation index *l* noted below each, and (c) their operation outputs, with the RMSE noted below each. (d) Test input pair of characteristic patterns and its corresponding (e) ground truths of the 16 operations, with the operation index *l* noted below each, and (f) their operation outputs, with the RMSE noted below each. Scale bars in (a) and (d) are 1 mm, indicating the physical scale after the upsampling process.

composed of random patterns. The second pair, shown in Fig. 4(d), was composed of characteristic patterns. Ground truths of their 16 SIMD logic operations are displayed in Figs. 4(b) and 4(e), respectively. The operation outputs are shown in Figs. 4(c) and 4(f), respectively, where all operations were successful, and their RMSEs were zero. Furthermore, the RMSEs for 256 test input pairs of random patterns were also found to be zero. These outcomes underscore the promising potential of DC. More detailed discussions are provided in the next section and the appendix.

4 Analysis

We conducted numerical analyses of the performance of DC under various optical conditions. Throughout this analysis, the experimental conditions were consistent with those described in

Sec. 3, except where otherwise noted. Further analyses are provided in Appendix A.

4.1 Number of DOEs

The computational performance of DC, with the number of DOEs set to $K-2$, was evaluated using the RMSEs, as illustrated in Fig. 5. In this evaluation, the number of input parallel bits N was set to 4, 16, 64, 128, and 256. Correspondingly, the upsampling factors along the x and y axes were adjusted to (64, 64), (32, 32), (16, 16), (8, 16), and (8, 8) $[=(s_x, s_y)]$, respectively, aiming to maintain consistent pixel counts of 128 $(=s_xN_x, s_yN_y)$ on the input layer after upsampling. The layer index of the input layer K_{in} was set to $\lfloor (K-2)/2 \rfloor + 2$. When the number of DOEs was zero, only the illumination pattern was optimized.

As illustrated in Fig. 5, the calculation error decreased with an increase in the number of DOEs. Additionally, the necessary number of DOEs for achieving error-free calculation increased with the number of input parallel bits N, but at a rate less than proportional to N. This rate of increase was smaller than predicted in previous works,[32,38] indicating an advantage of DC in terms of scalability and integration capability through the use of spatially parallelized optical processes for logic operations.

4.2 Multiplexing Advantage

The DOEs implemented 16 logic operations in a multiplexed manner in DC, as shown in Table 1. We confirmed the computational errors under different numbers of DOEs, denoted as $K-2$, when the optical cascade was designed for single logic operations. The layer index of the input layer, K_{in}, was set to $\lfloor (K-2)/2 \rfloor + 2$. In Fig. 6, computational errors for AND, OR, NAND, and NOR operations, selected from the 16 operations, are shown. Results for all 16 logic operations are provided in Fig. 8. In most of these results, error-free or nearly error-free calculations for single logic operations were achieved when the number of DOEs was greater than 6. On the other hand, as shown in Fig. 5, the necessary number of DOEs for multiplexing 16 logic operations was 9, which is significantly less than 6×16, for error-free calculations. Furthermore, the computational errors for some logic operations, such as XNOR and NOR, were reduced by multiplexing all the logic

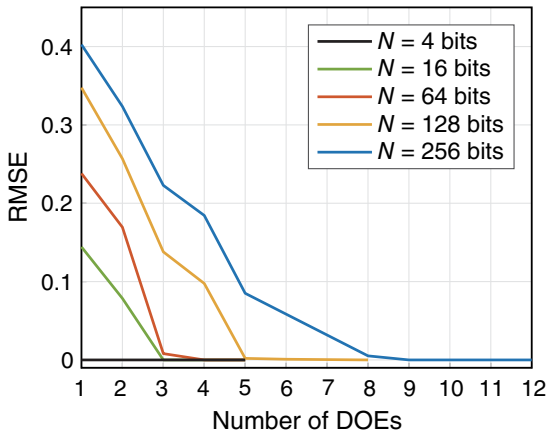

Fig. 5 Computational errors associated with the varying number of DOEs.

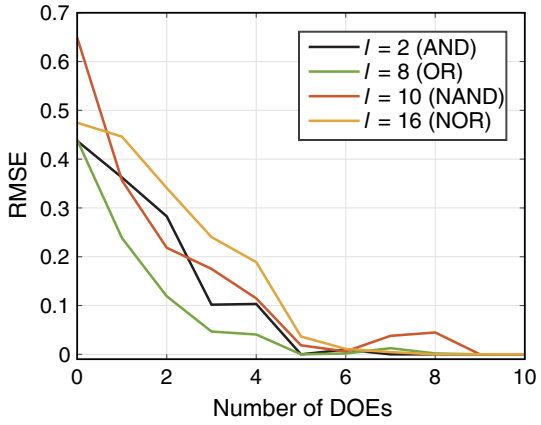

Fig. 6 Computational errors associated with the varying number of DOEs for single logic operations.

operations, which may help prevent falling into local minima in the optimization process. These results verified the advantage of multiplexing logic operations, in terms of both architecture configuration and training process, as well as the integration capability of DC.

5 Conclusion

We revived SC as DC by employing DNNs to achieve scalable and flexible optical SIMD operations. The optical cascade of DC consisted of reconfigurable illumination, DOEs, and an input layer. The illumination patterns and DOEs were designed to perform 16 logic operations on any binary input image pair, and the output intensity of the optical cascade was binarized to produce the final results. In this study, we achieved 16 switchable logic operations on 256 bits, which is an outstanding achievement compared with previous studies.[41–47] In contrast to these methods, where the optical diffraction processes from each spatial region of logic operations and bits must be separately designed, our method allows both interference between spatial regions of bits and interference between logic operations, enabling these regions to be densely located based on end-to-end designs of the illumination patterns and DOEs. This advantage has enabled high scalability and integration capability with all-optical operation, eliminating the need for computational encoding and decoding, all of which were numerically demonstrated.

An issue with DC for practical applications is its low energy efficiency. This may be addressed by adopting illumination with phase modulation and optimizing physical conditions, including the layer interval, buffer, and image sensor. We are planning the physical implementation of DC based on the setups shown in Sec. 7. Owing to its flexible and reconfigurable architecture offered by a learning-based approach, DC can be extended to a versatile range of inputs and operations beyond SIMD logic operations, such as image processing, including image filtering, and advanced reconfigurable optical computing methods like optical FPGAs. Furthermore, incorporating multiplexing in various optical quantities, such as time,[61,62] wavelength,[63] polarization,[39,64] and orbital angular momentum,[42] would enhance the computational capacity in DC. Thus, our study on DC offers a novel design architecture for optical computers and optical accelerators and paves the way for a next-generation optical computing paradigm.

6 Appendix A: Supplementary Analyses

This appendix provides detailed analyses of DC. Throughout this appendix, the experimental conditions are consistent with those described in Sec. 3, except where otherwise noted.

6.1 Training Process

Figure 7 illustrates the trends of the cost function based on the MSE without the binarization process in Eq. (15) and the computational error based on the RMSE in Eq. (25) during the training process. At the end of the training process, the cost function converged to nearly zero but not exactly zero. On the other hand, the computational error converged to exactly zero around the final iteration step. This indicates that the binarization process rectified the optical outputs and eliminated their small errors.

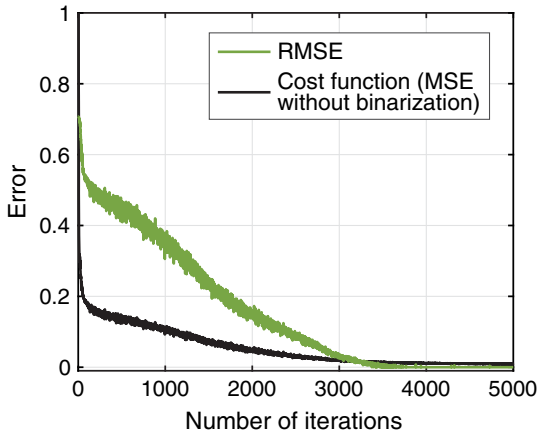

Fig. 7 Error trends during the training process.

The code for the numerical demonstrations in this study was implemented with MATLAB 2022a (MathWorks) and was executed on a computer with an AMD EPYC 7763 64-core processor at a clock rate of 2.45 GHz and an NVIDIA A100 SXM4 with 80 GB of memory. The computational time for the entire training process was ∼7 h.

6.2 Multiplexing Advantage

In Sec. 4.2, the computational errors of DC designed for single-logic operations of AND, OR, NAND, and NOR were selectively presented. The errors for all 16 logic operations are shown in Fig. 8. In most logic operations, error-free or nearly error-free calculations were achieved when the number of DOEs was greater than ∼6. On the other hand, multiplexing all 16 logic operations achieved error-free calculation when the number of DOEs was 9, as shown in Fig. 5, which was much smaller than 6×16 and supported the advantage of multiplexing the logic operations.

6.3 Physical Volume of DC

The physical volume of the optical cascade in Sec. 3.2 was calculated as $3.89 \times 10^{12}\lambda^3 (\approx 5.86 \times 10^{11}\ \mu\mathrm{m}^3)$, where the

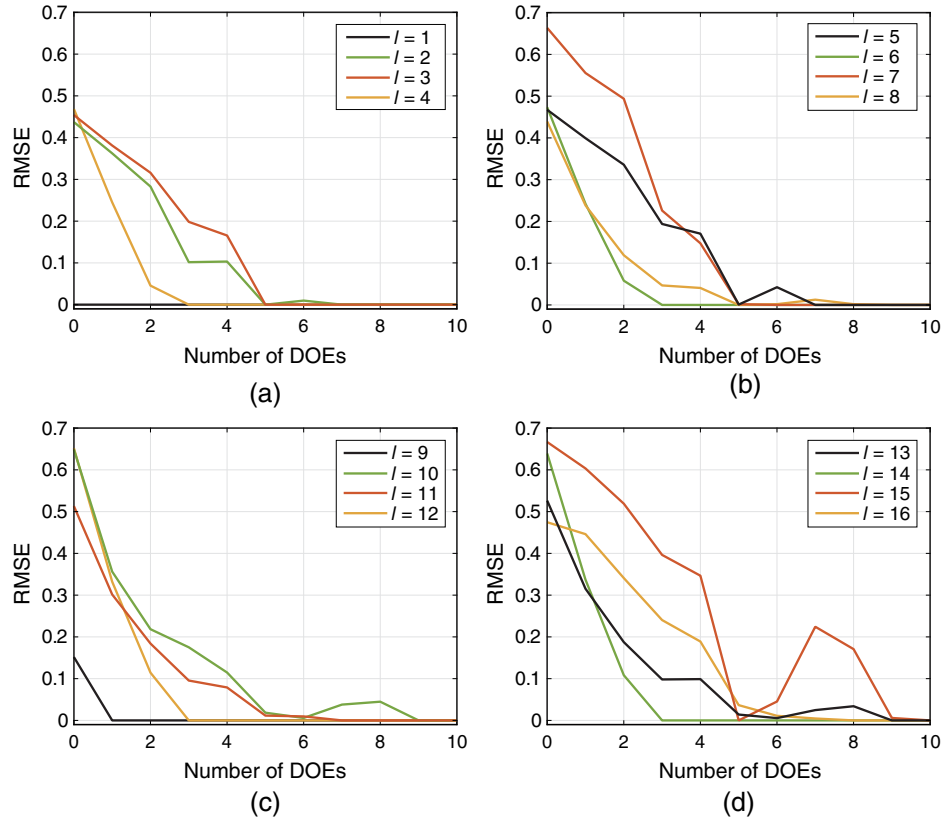

Fig. 8 Computational errors associated with the varying number of DOEs for single logic operations. (a) $1 \leq l \leq 4$, (b) $5 \leq l \leq 8$, (c) $9 \leq l \leq 12$, and (d) $13 \leq l \leq 16$.

pixel pitch on the DOEs was 16λ and the intervals between the layers in the optical cascade were $3 \times 10^4\lambda$, respectively. We investigated the computational error with respect to the reduction of the physical volume by scaling down both the pixel pitch and the interval with the same magnification ratio, varying the pixel pitch from $1/8\lambda$ to 16λ by powers of two. The result is shown in Fig. 9. In this case, the minimal physical volume without computational error was $5.94 \times 10^7\lambda^3$ ($\approx 8.94 \times 10^6$ μm^3), where the pixel pitch was λ ($= 0.532$ μm), and the interval was $1.17 \times 10^2\lambda$ ($\approx 6.23 \times 10$ μm). This result indicated that the minimal physical volume of DC is constrained by the diffraction limit.

6.4 Position of the Input Layer

The computational error was calculated by varying the position of the input layer K_{in} from 2 to 11 in the optical cascade with 11 layers, as depicted in Fig. 10. This result shows the importance of the DOEs downstream from the input layer in reducing computational error. It suggests that there is an advantage in positioning the input layer at an upper layer, excluding the top one.

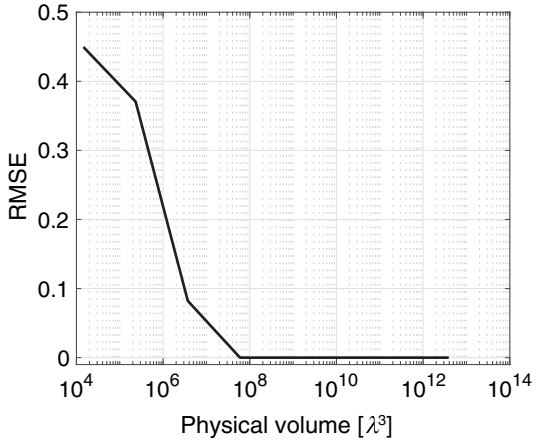

Fig. 9 Computational errors associated with the varying physical volume of DC.

6.5 Energy Efficiency

We evaluated the light-energy efficiency of DC using the following definition:

$$\text{Energy efficiency} = \frac{\sum_{\forall}[\mathcal{O}[|\boldsymbol{w}_{K+1}|^2]]|_{l=9,\boldsymbol{f}=1}}{P_x P_y}. \tag{26}$$

Here, the denominator represents the total input energy to the optical cascade. The numerator is the total energy on the output area of interest, calculated when the logic operation is configured to produce one output ($l = 9$) and all elements of the input pair $\boldsymbol{f}$ are set to 1. The energy efficiency was assessed with respect to the scaling factor a and the width of the buffer $\boldsymbol{t}$.

6.5.1 Scaling factor

The light-energy efficiency is associated with the scaling factor a, which amplifies or attenuates the signals captured by the image sensor before the binarization process. A larger a indicates lower energy efficiency and vice versa. In the above demonstrations and analyses, a was included in the optimized parameters, as shown in Eq. (20). Here, a was set to a specific value and was not updated during the optimization process. Once the optimizations of the illumination pattern $\boldsymbol{r}_l$ and the DOEs $\boldsymbol{\phi}_k$ were completed, the energy efficiency in Eq. (26) and the computational error were calculated. This process was repeated by changing a from 2 to 32. The relationship between the energy efficiency and the computational error is shown in Fig. 11. The RMSEs for the energy efficiencies between 1.81% and 8.71% were less than 2.38×10^{-3}. Therefore, nearly error-free calculation was achieved within this range of energy efficiencies.

The primary sources of energy loss were amplitude modulation on the illumination plane, light leakage from the optical cascade, and the cropping of the limited square area by the image sensor at the end of the optical cascade. The first issue can be solved by employing phase modulation on the illumination plane, although its modulation speed is lower than that of amplitude modulation on currently available SLMs. The second issue may be alleviated by reducing the intervals between layers. The third issue can be addressed by increasing the sensor area, employing anisotropic sampling, or utilizing anamorphic

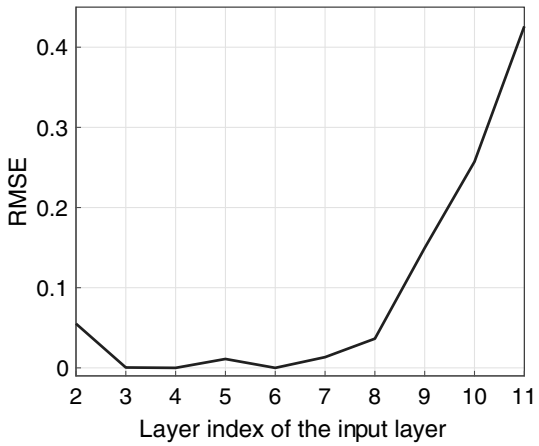

Fig. 10 Computational errors under different positions of the input layer.

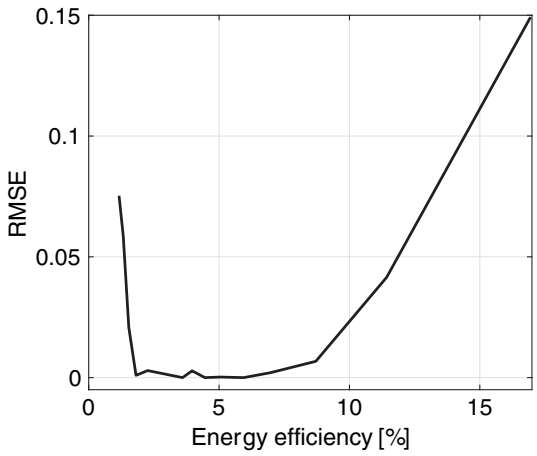

Fig. 11 Relationship between computational errors and energy efficiencies when varying the scaling factor.

imaging. Another approach to improve energy efficiency is to increase the width of the buffer, as indicated in the next section. The trade-off between energy efficiency and computational error will not be an issue in a proof-of-concept experimental demonstration with an optical setup, such as those shown in Sec. 7, by increasing the illumination intensity. However, this issue must be considered in practical applications, where energy efficiency in computation is a crucial factor.

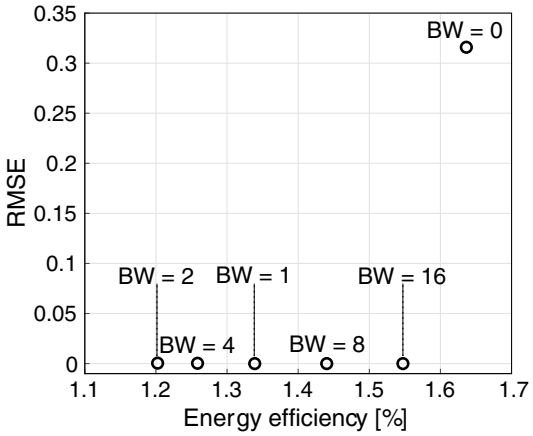

Fig. 12 Relationship between computational errors and energy efficiencies with the varying buffer width (BW [pixels]).

6.5.2 Buffer width

The buffer $\boldsymbol{t}$ was introduced into DC to compensate for the light intensities transmitted or blocked by the input pair. It was expected to eliminate the computational encoding and decoding processes employed in previous methods for optical logic operations, including the SC scheme. In the above demonstrations and analyses, the buffer width was set to 16 pixels. The plots in Fig. 12 show the energy efficiencies and computational errors at different buffer widths, including zero width. In this analysis, the scaling factor a was included in the optimization parameters. This result supported the necessity of the buffer for error-free calculation. Furthermore, a larger buffer width increased the energy efficiency.

The RMSE for the logic operations at $1 \leq l \leq 8$ without the buffer was reduced from 4.24×10^{-2} to 0 by adding a buffer of one pixel. On the other hand, the RMSE for the logic operations at $9 \leq l \leq 16$ with no buffer was significantly improved from 4.40×10^{-1} to 0 by adding a one-pixel buffer. As shown in Table 1, operations at $1 \leq l \leq 8$ do not include the operation with an input of $\boldsymbol{f}_{\text{left}} = 0$, $\boldsymbol{f}_{\text{right}} = 0$ and an output of $\hat{\boldsymbol{g}}_l = 1$. Conversely, the operations at $9 \leq l \leq 16$ include such an

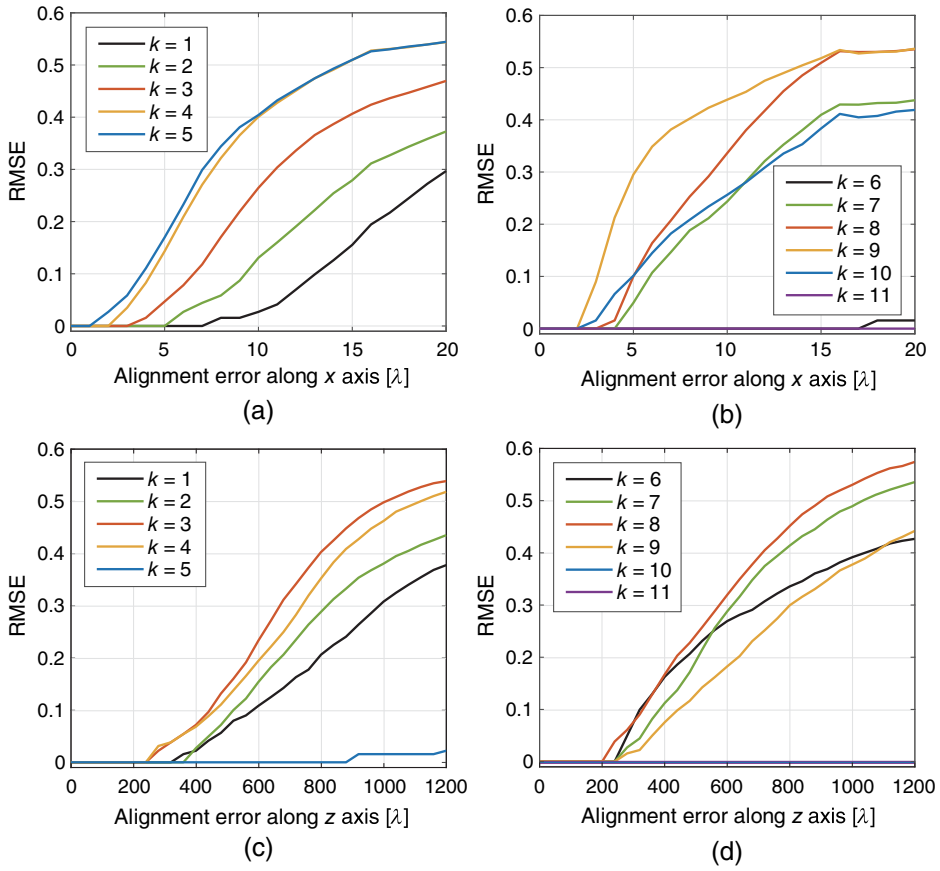

Fig. 13 Relationships between computational errors and alignment errors along the x axis on the layers (a) between the illumination and the one before the input ($1 \leq k \leq 5$), and (b) between the input and the end ($6 \leq k \leq 11$); and those along the z axis on the layers (c) between the illumination and the one before the input ($1 \leq k \leq 5$), and (d) between the input and the end ($6 \leq k \leq 11$).

operation. This result also verified the role of the buffer—compensating for the balance between the light intensities of the input and output in the optical cascade.

6.6 Alignment Error

An issue in the physical demonstration of DC will be alignment errors. The system's performance under alignment errors along the x and z axes is presented in Fig. 13, showing the computational error with one-dimensional alignment on the individual layers, including the illumination and input layers. The alignment error along the y axis was omitted in this analysis because its impact is considered similar to that along the x axis due to their symmetry. As shown in these results, the impact of the alignment error along the x axis was greater than that along the z axis.

Several methods for compensating for alignment errors in DNNs have been proposed, and these can be applied to configure our setup. The first approach is enhancing robustness against alignment errors or model errors by introducing them during the computational training process.[65] The second approach is using a closed-loop process to feed back alignment errors or model mismatches to controllable optical elements, such as SLMs.[66,67] The third approach is incorporating integrated chip fabrication techniques, which significantly reduce alignment errors or model mismatches.[32,43]

7 Appendix B: Optical Setup

Two candidate experimental setups for DC are presented in Fig. 14. Both setups employ DMDs for the reconfigurable illumination and input due to the high speed and high contrast in DMD modulation. The first setup, shown in Fig. 14(a), uses transmissive phase modulation elements, such as DOEs. While it can be bulky, it is easier to align optical components. The polarization state of the light may optionally be controlled to prevent stray light from the DMDs. This type of setup has been demonstrated in DNNs for object classification and classical or quantum logic gates.[28,31,41,46] The second setup, shown in Fig. 14(b), uses a mirror and reflective phase modulation elements, such as SLMs. This reflective setup may be more compact and compatible with the closed-loop approach described in Sec. 6.6 to compensate for alignment errors. The diagonal propagation process can be considered when designing the phase modulations using rotational transformation in numerical diffraction.[68] This type of reflective setup has been demonstrated in DNNs for a beam mode converter, a quantum gate, and optical reservoir computing.[69–71]

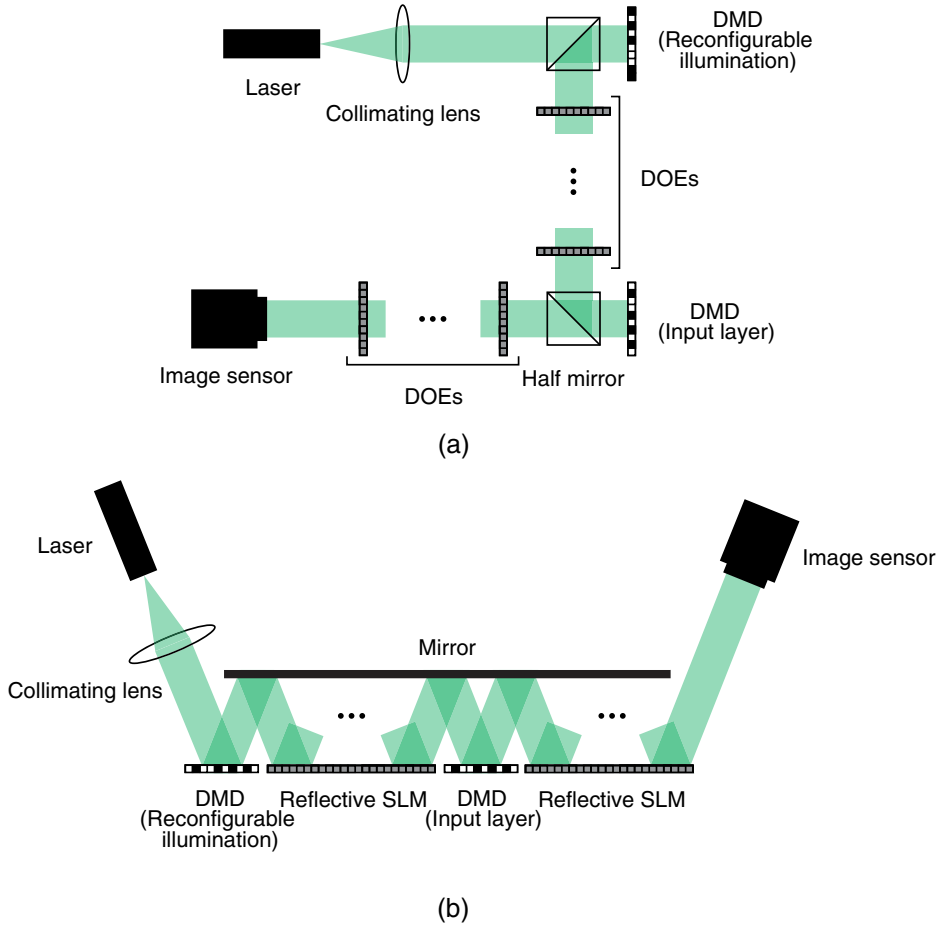

Fig. 14 Candidates for the experimental setups of DC using (a) transmissive phase modulation with DOEs and (b) reflective phase modulation with SLMs.

Disclosures

The authors declare no conflicts of interest.

Code and Data Availability

Data may be obtained from the authors upon reasonable request.

Acknowledgments

We gratefully acknowledge the insightful and thoughtful leadership of the late Dr. Makoto Naruse and extend our deepest condolences. We also thank Mr. Yuya Hidaka at The University of Tokyo for fruitful discussions. This work was supported by Japan Society for the Promotion of Science (Grant Nos. JP20K05361, JP22H05197, and JP23K26567).

References

1. G. E. Moore, "Cramming more components onto integrated circuits," *Proc. IEEE* **86**(1), 82–85 (1998).
2. H. J. Caulfield and S. Dolev, "Why future supercomputing requires optics," *Nat. Photonics* **4**(5), 261–263 (2010).
3. K. Kitayama et al., "Novel frontier of photonics for data processing—photonic accelerator," *APL Photonics* **4**(9), 090901 (2019).
4. R. P. Bocker, "Matrix multiplication using incoherent optical techniques," *Appl. Opt.* **13**(7), 1670–1676 (1974).
5. J. W. Goodman, A. Dias, and L. Woody, "Fully parallel, high-speed incoherent optical method for performing discrete Fourier transforms," *Opt. Lett.* **2**(1), 1–3 (1978).
6. R. A. Athale and W. C. Collins, "Optical matrix–matrix multiplier based on outer product decomposition," *Appl. Opt.* **21**(12), 2089–2090 (1982).
7. M. Gruber, J. Jahns, and S. Sinzinger, "Planar-integrated optical vector-matrix multiplier," *Appl. Opt.* **39**(29), 5367–5373 (2000).
8. Y. Owechko et al., "Holographic associative memory with nonlinearities in the correlation domain," *Appl. Opt.* **26**(10), 1900–1910 (1987).
9. E. G. Paek and D. Psaltis, "Optical associative memory using Fourier transform holograms," *Opt. Eng.* **26**(5), 428–433 (1987).
10. M. Ishikawa et al., "Optical associatron: a simple model for optical associative memory," *Appl. Opt.* **28**(2), 291–301 (1989).
11. J. Tanida and Y. Ichioka, "Optical logic array processor using shadowgrams," *J. Opt. Soc. Am.* **73**(6), 800–809 (1983).
12. Y. Ichioka and J. Tanida, "Optical parallel logic gates using a shadow-casting system for optical digital computing," *Proc. IEEE* **72**(7), 787–801 (1984).
13. J. Tanida and Y. Ichioka, "OPALS: optical parallel array logic system," *Appl. Opt.* **25**(10), 1565–1570 (1986).
14. K.-H. Brenner, A. Huang, and N. Streibl, "Digital optical computing with symbolic substitution," *Appl. Opt.* **25**(18), 3054–3060 (1986).
15. D. A. Miller, "Self-configuring universal linear optical component," *Photonics Res.* **1**(1), 1–15 (2013).
16. Y. Shen et al., "Deep learning with coherent nanophotonic circuits," *Nat. Photonics* **11**(7), 441–446 (2017).
17. N. C. Harris et al., "Linear programmable nanophotonic processors," *Optica* **5**(12), 1623–1631 (2018).
18. M. Zhang, L. Wang, and P. Ye, "All optical XOR logic gates: technologies and experiment demonstrations," *IEEE Commun. Mag.* **43**(5), S19–S24 (2005).
19. Y. Zhang, Y. Zhang, and B. Li, "Optical switches and logic gates based on self-collimated beams in two-dimensional photonic crystals," *Opt. Express* **15**(15), 9287–9292 (2007).
20. Y.-D. Wu, T.-T. Shih, and M.-H. Chen, "New all-optical logic gates based on the local nonlinear Mach-Zehnder interferometer," *Opt. Express* **16**(1), 248–257 (2008).
21. K.-S. Lee and S.-K. Kim, "Conceptual design of spin wave logic gates based on a Mach–Zehnder-type spin wave interferometer for universal logic functions," *J. Appl. Phys.* **104**(5), 053909 (2008).
22. J. Dong, X. Zhang, and D. Huang, "A proposal for two-input arbitrary Boolean logic gates using single semiconductor optical amplifier by picosecond pulse injection," *Opt. Express* **17**(10), 7725–7730 (2009).
23. Y. Fu, X. Hu, and Q. Gong, "Silicon photonic crystal all-optical logic gates," *Phys. Lett. A* **377**(3–4), 329–333 (2013).
24. D. G. Sankar Rao, S. Swarnakar, and S. Kumar, "Performance analysis of all-optical NAND, NOR, and XNOR logic gates using photonic crystal waveguide for optical computing applications," *Opt. Eng.* **59**(5), 057101 (2020).
25. Y. Xie et al., "Programmable optical processor chips: toward photonic RF filters with DSP-level flexibility and MHz-band selectivity," *Nanophotonics* **7**(2), 421–454 (2017).
26. W. Bogaerts et al., "Programmable photonic circuits," *Nature* **586**(7828), 207–216 (2020).
27. Z. Ying et al., "Electronic-photonic arithmetic logic unit for high-speed computing," *Nat. Commun.* **11**(1), 2154 (2020).
28. X. Lin et al., "All-optical machine learning using diffractive deep neural networks," *Science* **361**(6406), 1004–1008 (2018).
29. T. Yan et al., "Fourier-space diffractive deep neural network," *Phys. Rev. Lett.* **123**(1), 023901 (2019).
30. J. Weng et al., "Meta-neural-network for real-time and passive deep-learning-based object recognition," *Nat. Commun.* **11**(1), 6309 (2020).
31. H. Chen et al., "Diffractive deep neural networks at visible wavelengths," *Engineering* **7**(10), 1483–1491 (2021).
32. H. Zhu et al., "Space-efficient optical computing with an integrated chip diffractive neural network," *Nat. Commun.* **13**(1), 1044 (2022).
33. C. Qian et al., "Dynamic recognition and mirage using neuro-metamaterials," *Nat. Commun.* **13**(1), 2694 (2022).
34. Y. Luo et al., "Computational imaging without a computer: seeing through random diffusers at the speed of light," *elight* **2**(1), 4 (2022).
35. Ç. Işıl et al., "Super-resolution image display using diffractive decoders," *Sci. Adv.* **8**(48), eadd3433 (2022).
36. M. Huang et al., "Diffraction neural network for multi-source information of arrival sensing," *Laser Photonics Rev.* **17**(10), 2300202 (2023).
37. T. Igarashi, M. Naruse, and R. Horisaki, "Incoherent diffractive optical elements for extendable field-of-view imaging," *Opt. Express* **31**(19), 31369–31382 (2023).
38. O. Kulce et al., "All-optical information-processing capacity of diffractive surfaces," *Light Sci. Appl.* **10**(1), 25 (2021).
39. J. Li et al., "Polarization multiplexed diffractive computing: all-optical implementation of a group of linear transformations through a polarization-encoded diffractive network," *Light Sci. Appl.* **11**(1), 153 (2022).
40. J. Li et al., "Massively parallel universal linear transformations using a wavelength-multiplexed diffractive optical network," *Adv. Photonics* **5**(1), 016003 (2023).
41. C. Qian et al., "Performing optical logic operations by a diffractive neural network," *Light Sci. Appl.* **9**(1), 59 (2020).
42. P. Wang et al., "Orbital angular momentum mode logical operation using optical diffractive neural network," *Photonics Res.* **9**(10), 2116–2124 (2021).
43. S. Zarei and A. Khavasi, "Realization of optical logic gates using on-chip diffractive optical neural networks," *Sci. Rep.* **12**(1), 15747 (2022).
44. Y. Luo, D. Mengu, and A. Ozcan, "Cascadable all-optical NAND gates using diffractive networks," *Sci. Rep.* **12**(1), 7121 (2022).
45. X. Liu et al., "Parallelized and cascadable optical logic operations by few-layer diffractive optical neural network," *Photonics* **10**(5), 503 (2023).

46. X. Ding et al., "Metasurface-based optical logic operators driven by diffractive neural networks," *Adv. Mater.* **36**(9), 2308993 (2024).
47. X. Lin et al., "Polarization-based all-optical logic gates using diffractive neural networks," *J. Opt.* **26**(3), 035701 (2024).
48. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, Springer (2013).
49. J. A. Dev, "Bitcoin mining acceleration and performance quantification," in *IEEE 27th Can. Conf. Electr. and Comput. Eng. (CCECE)*, IEEE, pp. 1–6 (2014).
50. A. Reuther et al., "Survey and benchmarking of machine learning accelerators," in *High Perform. Extreme Comput. Conf. (HPEC)*, IEEE, pp. 1–9 (2019).
51. J. D. Owens et al., "GPU computing," *Proc. IEEE* **96**(5), 879–899 (2008).
52. N. P. Jouppi et al., "In-datacenter performance analysis of a tensor processing unit," in *Proc. 44th Annu. Int. Symp. Comput. Architect.*, IEEE, pp. 1–12 (2017).
53. J. Rose, A. El Gamal, and A. Sangiovanni-Vincentelli, "Architecture of field-programmable gate arrays," *Proc. IEEE* **81**(7), 1013–1029 (1993).
54. X. Zhang et al., "Machine learning on FPGAs to face the IoT revolution," in *IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, IEEE, pp. 894–901 (2017).
55. K. R. Hari et al., "Cryptocurrency mining–transition to cloud," *Int. J. Adv. Comput. Sci. Appl.* **6**(9), 115–124 (2015).
56. E. Nurvitadhi et al., "Accelerating recurrent neural networks in analytics servers: comparison of FPGA, CPU, GPU, and ASIC," in *26th Int. Conf. Field Programm. Logic and Appl. (FPL)*, pp. 1–4 (2016).
57. J. W. Goodman, *Introduction to Fourier Optics*, Roberts and Company (2005).
58. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv:1412.6980 (2014).
59. M. Courbariaux, Y. Bengio, and J.-P. David, "BinaryConnect: training deep neural networks with binary weights during propagations," arXiv:1511.00363 (2015).
60. H. Wang et al., "Toward near-perfect diffractive optical elements via nanoscale 3D printing," *ACS Nano* **14**(8), 10452–10461 (2020).
61. Z. Zhang et al., "Space-time projection enabled ultrafast all-optical diffractive neural network," *Laser and Photonics Rev.* **18**, 2301367 (2024).
62. J. Zhou, H. Pu, and J. Yan, "Spatiotemporal diffractive deep neural networks," *Opt. Express* **32**(2), 1864–1877 (2024).
63. Z. Duan, H. Chen, and X. Lin, "Optical multi-task learning using multi-wavelength diffractive deep neural networks," *Nanophotonics* **12**(5), 893–903 (2023).
64. X. Luo et al., "Metasurface-enabled on-chip multiplexed diffractive neural networks in the visible," *Light Sci. Appl.* **11**(1), 158 (2022).
65. D. Mengu et al., "Misalignment resilient diffractive optical networks," *Nanophotonics* **9**(13), 4207–4219 (2020).
66. M. Nakajima et al., "Physical deep learning with biologically inspired training method: gradient-free approach for physical hardware," *Nat. Commun.* **13**(1), 7847 (2022).
67. A. Momeni et al., "Training of physical neural networks," arXiv:2406.03372 (2024).
68. K. Matsushima, "Formulation of the rotational transformation of wave fields and their application to digital holography," *Appl. Opt.* **47**(19), D110–D116 (2008).
69. N. K. Fontaine et al., "Laguerre-Gaussian mode sorter," *Nat. Commun.* **10**(1), 1865 (2019).
70. Q. Wang et al., "Ultrahigh-fidelity spatial mode quantum gates in high-dimensional space by diffractive deep neural networks," *Light Sci. Appl.* **13**(1), 10 (2024).
71. M. Yildirim et al., "Nonlinear processing with linear optics," *Nat. Photonics* 1–7 (2024).

Ryosuke Mashiko is a graduate student in the Department of Information Physics and Computing at the Graduate School of Information Science and Technology, the University of Tokyo, Tokyo, Japan. He received his BE degree in engineering from the University of Tokyo, in 2023. His work focuses on the development of photonic computing and computational imaging.

Makoto Naruse was a professor in the Department of Information Physics and Computing, Graduate School of Information Science and Technology, the University of Tokyo, Tokyo, Japan. He received his BE, ME, and PhD degrees in engineering from the University of Tokyo, in 1994, 1996, and 1999, respectively.

Ryoichi Horisaki is an associate professor in the Department of Information Physics and Computing at the Graduate School of Information Science and Technology, the University of Tokyo, Tokyo, Japan. He received his BS degree in engineering from Nara National College of Technology in 2005, and his MS and PhD degrees in information science from Osaka University in 2007 and 2010, respectively. His research interests include information photonics and computational imaging.