Research article

Takashi Asano* and Susumu Noda

Iterative optimization of photonic crystal nanocavity designs by using deep neural networks

https://doi.org/10.1515/nanoph-2019-0308
Received August 10, 2019; revised October 12, 2019; accepted October 22, 2019

Abstract: Devices based on two-dimensional photonic-crystal nanocavities, which are defined by their air hole patterns, usually require a high quality (*Q*) factor to achieve high performance. We demonstrate that hole patterns with very high *Q* factors can be efficiently found by the iteration procedure consisting of machine learning of the relation between the hole pattern and the corresponding *Q* factor and new dataset generation based on the regression function obtained by machine learning. First, a dataset comprising randomly generated cavity structures and their first principles *Q* factors is prepared. Then a deep neural network is trained using the initial dataset to obtain a regression function that approximately predicts the *Q* factors from the structural parameters. Several candidates for higher *Q* factors are chosen by searching the parameter space using the regression function. After adding these new structures and their first principles *Q* factors to the training dataset, the above process is repeated. As an example, a standard silicon-based L3 cavity is optimized by this method. A cavity design with a high *Q* factor exceeding 11 million is found within 101 iteration steps and a total of 8070 cavity structures. This theoretical *Q* factor is more than twice the previously reported record values of the cavity designs detected by the evolutionary algorithm and the leaky mode visualization method. It is found that structures with higher *Q* factors can be detected within less iteration steps by exploring not only the parameter space near the present highest-*Q* structure but also that distant from the present dataset.

Keywords: optimization; neural network; photonic crystals; nanocavities; *Q* factor.

***Corresponding author: Takashi Asano,** Department of Electronic Science and Engineering, Kyoto University, Kyoto 615-8510, Japan, e-mail: tasano@kuee.kyoto-u.ac.jp.
https://orcid.org/0000-0003-1456-1995
Susumu Noda: Department of Electronic Science and Engineering, Kyoto University, Kyoto 615-8510, Japan

1 Introduction

Photonic nanocavities based on artificial defects in two-dimensional (2D) photonic-crystal (PC) slabs [1–11] have received significant attention as structures that enable preservation of photons for extended times in small modal volumes. 2D-PC slab cavities are usually defined by defects in the triangular air hole lattice of the PC. For example, cavities can be defined by a defect consisting of three missing air holes (the so-called L3 cavity), a single missing hole (H0 cavity), or a line defect with a modulation of the lattice constants (heterostructure cavity). Photons of the cavity modes are confined in such nanocavities in the in-plane and vertical directions by Bragg reflection due to the air hole pattern of the 2D PC and total internal reflection due to the refractive index contrast between the PC slab and the surrounding air or cladding layers, respectively. We note that the in-plane reflection is usually almost perfect, while the vertical reflection is only partial [2]. Thus, the total spectral intensity of the wavevector components that do not fulfill the total internal reflection condition, i.e. the leaky components, determines the cavity's quality (*Q*) factor [12]. So far, various methods of optimizing cavity designs with respect to the *Q* factor have been proposed and demonstrated [2–5, 12–19]. Among them, the Gaussian envelope approaches [2, 3], the leaky position visualization approach [17], and the analytic inverse problem approaches [13, 14] utilize the knowledge of the physics of photon confinement mentioned above. For instance, the analytic inverse problem approaches are based on approximations that relate the cavities' structural parameters to the mode fields and thus allow us to explicitly determine an optimized cavity geometry with less leaky components [13, 14]. This type of approaches is very useful to optimize specific structural parameters, but targets are limited because suited analytical expressions are only available for certain cavity types. On the other hand, the Gaussian envelope and leaky position visualization approaches improve cavity designs based on the differences between the mode field calculated for the actual structure and the ideal mode field, which is artificially generated and has a minimum of leaky components

[2, 3, 17]. The comparison of these fields enables identification of spatial positions where leakage of photons occurs. However, since these approaches cannot predict the optimized structure, the modifications required for a reduction of leakage have to be manually identified by trial and error. While these approaches are useful in early optimization stages, they cannot utilize the large degree of freedom that is inherent to the 2D geometry of the air hole pattern. The reports on optimization of 2D-PC nanocavity designs by these approaches have so far considered only up to nine structural parameters (e.g. symmetric displacements of certain holes) for optimization [2, 3, 17], because it is difficult to manually locate better air hole patterns in the high-dimensional parameter space consisting of the positions of all individual air holes. Obviously, more systematic and automated methods of exploring high-dimensional parameter spaces are required to fully utilize the potential of 2D-PC nanocavities.

The adjoint method has proven very effective in the optimization of nanophotonic devices such as demultiplexers, grating couplers, and waveguide bends [20], in which the emphasis lies in optimization of transmission properties. While there are also reports that use the adjoint method for optimizing designs of ring resonators and cavities with respect to the Q factor in 1D and 2D calculations [21–24], 3D calculations are inevitable to evaluate the Q factors of 2D-PC nanocavity designs in the high-Q region. However, the Q factors that have been obtained in such 3D adjoint-method calculations are relatively small ($\sim 1\times10^5$) [25, 26] compared to those achieved by the methods explained in the following ($>1\times10^6$). Minkov et al. utilized a genetic algorithm to explore the parameter space of the 2D-PC air hole pattern and succeeded in tuning up to 11 parameters to find more suited nanocavity structures without using the physical knowledge of leaky components [15, 16, 18]. However, this approach requires a relatively large number of randomly generated sample cavity structures and their calculated Q factors: they have reported that 100 cycles$\times$80 individuals$=$8000 sample cavities (300 cycles$\times$120 individuals$=$36,000 sample cavities) were required to optimize five (seven) parameters in the L3 (H0) cavity [16]. The relatively large number of required sample cavities is considered to be a consequence of the genetic algorithm, which basically utilizes only the good cavities among the sample cavities generated in each cycle. Recently, we have proposed an approach based on deep learning, demonstrating optimization of 27 parameters of a heterostructure cavity using a training dataset consisting of 1000 randomly generated air hole patterns and their calculated Q factors [19]. In [19], we trained a neural network (NN) by the sample dataset to obtain an approximate function of the Q factor with respect to the structural parameters. This regression function was then employed to detect new cavity structures that are likely to exhibit higher Q factors. The important point is that not only high-Q structures but also moderate or low-Q structures can be useful when searching new cavity geometries with higher Q factors (since both improve the accuracy of the regression function developed by the NN), although high-Q sample cavity structures are of course more helpful. However, one problem of this approach is that structures with Q factors much higher than that of the base cavity design are rarely generated during the random preparation of the training dataset. Therefore, the accuracy of the regression function at the parameter space near extremely high Q factors is low.

In this report, we propose an iterative optimization method to overcome this problem: here, the candidate structures for higher-Q factors identified by the regression function at the present iteration step are added to the training dataset for the next step. The new dataset is used to derive an improved regression function. To increase the diversity of the new candidates, several different candidate-selection constraints are defined, and their combinations are used to efficiently explore the parameter space. In order to avoid strong influences of initial discoveries, one constraint is that the new candidate should lie at a parameter space distant from the structures that have already been analyzed. Additionally, we employ several NNs that learn the dataset in different orders, resulting in different regression functions. With these we can partly account for the uncertainty of the prediction by a NN. By repeating the optimization cycles, cavity structures that are important for detection of high-Q cavity structures are automatically accumulated in the dataset. To demonstrate this, we optimize the design of a silicon (Si) L3 cavity via 25 parameters. We are able to detect a structure with a maximum Q factor of almost 11 million by generating a total of 8070 sample structures within 101 iterations. This theoretical Q factor is more than two times larger than the Q factors of Si-based L3 cavity structures found by the genetic algorithm [16] and leaky mode visualization approaches [17].

2 Framework

In this section we explain the procedures of the proposed iterative optimization method, which contains the preparation, learning, structure search, validation, and dataset update phases. The latter four phases are repeated to iteratively improve the regression function developed in the learning phase and the following structure search. The general design of the preparation and learning phases can be found in [19]. First of all, we assume that the type of 2D-PC cavity that is to be optimized is known (in Section 3 we choose the L3 cavity). Next, the preparation phase

consisting of the following three procedures (I)–(III) has to be implemented:

I. Select the structural parameters of the base cavity (such as air hole positions and radii) that should be considered for optimization. Generate many sample cavity structures by randomly varying the selected parameters within a certain meaningful range.
II. Calculate the Q factors of the sample cavities generated in (I) by a first principles method to obtain the training dataset consisting of the sample cavity structures and the corresponding Q factors.
III. Prepare deep NNs that have input nodes corresponding to the structural parameters selected in (I) and have a single output node corresponding to the Q factor.

The learning phase is described by the following procedure:

IV. Train the deep NNs prepared in (III) to learn the relation between the structure and the Q factor using the dataset prepared in (I) and (II) (only for the first round) or the updated dataset obtained in (VII) (for the following rounds). Let each deep NN learn the dataset in a different order so that they acquire different approximation functions of Q. To avoid memory effects, the NN's weights are reset at the beginning of each iteration cycle.

The structure search phase consists of the following procedure:

V. Starting from a randomly chosen initial cavity structure, gradually change the structural parameters using the gradient (in the parameter space) of the approximated Q factor that is predicted by a trained deep NN. By this process, one new candidate structure with a potentially higher Q factor is located. Various candidate structures are prepared by using different deep NNs and by applying different constraints (described later).

The validation phase is straightforward:

VI. Determine the accurate Q factors of the candidate structures by a first principles calculation.

After the learning, structure search, and validation phases, the training dataset is updated and the next iteration cycle is carried out as follows:

VII. Add the sets of the structures obtained in (V) and the Q factors calculated in (VI) to the training dataset.
VIII. Go to (IV).

By repeating the procedures (IV)–(VII), the sample cavities that are important for locating high-Q structures are automatically accumulated, because both correct and wrong predictions constitute important information for the development of an improved regression function. Figure 1 briefly illustrates the concept of the approach for optimization explained above.

3 Optimization of the cavity design for a Si-based L3 nanocavity

In this section, we demonstrate the optimization of the cavity design for a L3 cavity made of Si by the proposed iterative optimization. The results are useful for device development and also provide a benchmark for the optimization performance of the presently used algorithm. The numbers given can be compared with those in previously reported methods [16, 17], because the Si-based L3 nanocavity is a standard 2D-PC nanocavity.

3.1 Preparation phase

Figure 2 shows the basic structure of the presently considered L3 nanocavity, where the lattice constant is a, the

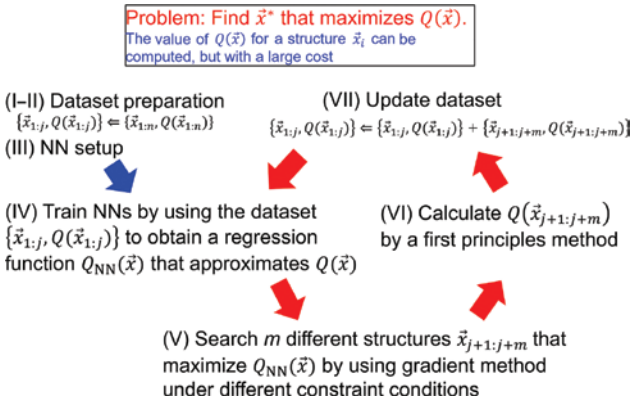

Figure 1: The iterative optimization proposed in this paper. Each cavity with the structural parameters selected in the preparation phase is represented by a unique high-dimensional vector $\vec{x}$. The accurate Q as a function of $\vec{x}$, $Q(\vec{x})$, can be calculated by first principle approaches but is costly to compute. $\vec{x}_{i:j}$ denotes the set of structures and consists of $\vec{x}_i$, $\vec{x}_{i+1}$, …, $\vec{x}_j$. $Q(\vec{x}_{i:j})$ denotes the corresponding set of Q factors, and $\{\vec{x}_{1:j}, Q(\vec{x}_{1:j})\}$ is used to refer to the dataset consisting of the chosen cavity structures and their Q factors. The number of sample structures in the initial dataset is n. $Q_{NN}(\vec{x})$ represents a low-cost regression function that approximates $Q(\vec{x})$ and is obtained by training a neural network (NN) using the training dataset $\{\vec{x}_{1:j}, Q(\vec{x}_{1:j})\}$. $Q_{NN}(\vec{x})$ is only used to locate new structures via the gradient, but the values are not explicitly discussed in this work.

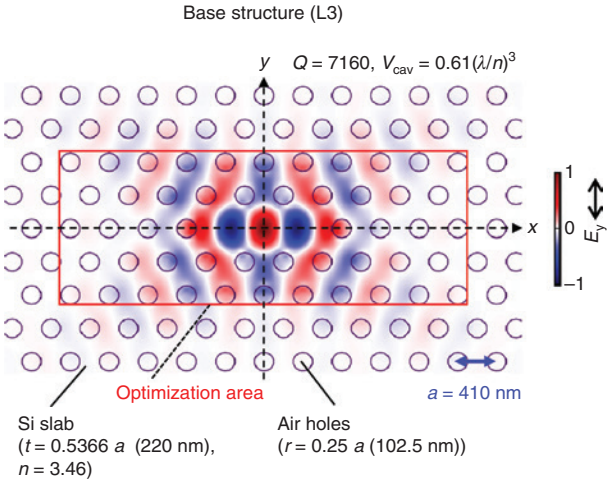

Figure 2: A three-missing-air-holes (L3) cavity is used as the base structure for structural optimization.
The lattice constant a is 410 nm. The circles indicate the air holes (hole radius: 102.5 nm = 0.25 a) formed in Si slab with a refractive index $n = 3.46$ and a thickness of 220 nm (0.5366 a). The distribution of the y component of the electric field (E_y) of the fundamental resonant mode is plotted in color. The theoretical Q factor and modal volume V_{cav} of the base structure determined by FDTD are 7160 and 0.61 $(\lambda/\eta)^3$, respectively. The displacements of the 50 air holes inside the red square are the structural parameters that are used to optimize the cavity design with respect to Q.

radius of each air hole is 0.25 a, the thickness of the slab is 0.5366 a, and the refractive index of the slab material (Si) is $n = 3.46$. These values were chosen by considering the standard dimensions of fabricated nanocavities ($a = 410$ nm, $t = 220$ nm) operating at optical communication wavelengths [9, 27] and the refractive index of Si at these wavelengths. The radii of the air holes are the same as those used in [16], and the slab thickness is similar to that in [16] (0.55 a). The color plot in Figure 1 shows the electric field distribution of the fundamental mode in the y direction (E_y). The distribution was calculated for the base cavity structure by a first principle method [three-dimensional finite-difference time-domain (3D-FDTD) method], and the resulting Q factor of the base structure is 7160. The modal volume V_{cav} of the mode is 0.61 cubic wavelengths in the material $(\lambda/n)^3$. Further details of the calculation conditions are provided in [19].

[Step (I)]: The positions of the 50 air holes within the area of 11 $(a) \times 5$ (rows) (indicated by the red square in Figure 2) are the structural parameters that are used to optimize the cavity design with respect to the Q factor, because most of the electric field intensity of the mode concentrates in this area [19]. We used the air hole positions as the variable parameters in the optimization process, because a specified but small offset in the air hole radius or shape that is different for each hole is more difficult to control in the etching step during fabrication. In contrast, the air hole positions, which are precisely defined in the electron beam writing process, can be reproduced with higher accuracy during etching. Each sample cavity structure (labelled by index i) is defined by the base structure and a set of 2D displacement vectors $\{\vec{d}_1, \vec{d}_2, \ldots\}_i$, where $\vec{d}_h = (d_{hx}, d_{hy})$ defines the displacement of the hth air hole in the x-y plane and h enumerates all air holes that are selected for structural optimization (from 1 to 50 in the present case). The parameter space vector of structure i, $\vec{x}_i$ as defined in Figure 1, is a single column vector with the structure $(d_{1x}, d_{1y}, d_{2x}, \ldots, d_{50y})^T$ and contains displacements corresponding to the single set $\{\vec{d}_1, \vec{d}_2, \ldots, \vec{d}_{50}\}_i$. Although we have 100 degrees of freedom in the 2D displacements of 50 air holes, the actual degrees of freedom in the present analysis are 25 because we have to impose mirror symmetries with respect to the central x and y axes to obtain high Q factors [12].

[Step (II)]: Random displacements are applied to all air holes in the x and y directions in such a way that the mirror symmetries of the structure are maintained and that a uniform distribution between $-0.1\,a$ and $0.1\,a$ is obtained. The appropriate magnitude of the fluctuation has been determined in previous manual optimizations of L3 cavities [2, 17]. In this demonstration, we initially prepare $n = 1000$ random nanocavity structures (the whole set is denoted by $\vec{x}_{1:n}$) using the above outlined displacement restrictions and calculate their Q factors using the 3D-FDTD method. The FDTD cell dimensions used in this work are about $a/10$ in x and y directions. For the discretization of the distribution of the dielectric constant in the FDTD calculation, we employed a sub-cell size of about $a/4000$, and the dielectric constants of each cell was determined by averaging over its sub-cells. Therefore, a change on the order of $a/4000$ in the dielectric constant distribution (which reflects the air hole displacements) can be resolved in the FDTD calculation of the Q factor. The obtained set of Q values, $Q(\vec{x}_{1:n})$, exhibits a distribution between 10^3 and 10^5, and the average is 6700 (see Figure 7A). Because the first principles Q values of the initial set are spread over two orders of magnitudes, and this difference should increase in the subsequent optimization cycles, we employ $\log_{10}(Q(\vec{x}_i))$ as the target of machine learning. As a result, the initial training dataset consists of the structural parameters $\vec{x}_i$ and $\log_{10}Q(\vec{x}_i)$ of the 1000 structures, i.e. $\{\vec{x}_{1:n}, \log_{10}Q(\vec{x}_{1:n})\}$ instead of $\{\vec{x}_{1:n}, Q(\vec{x}_{1:n})\}$.

[Step (III)]: Ten four-layer-NNs with the same configuration as in [19] are prepared (Figure 3). The input nodes are two-channel 2D tables, where each channel

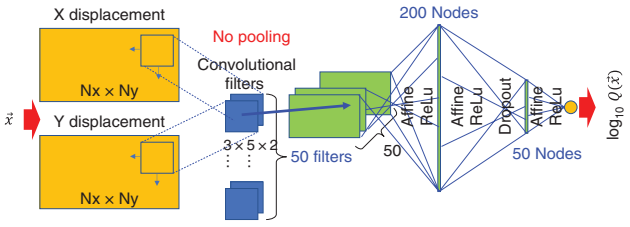

Figure 3: Configuration of the neural network prepared to learn the relationship between displacements of air holes and Q factors (ReLU: rectified linear unit. Affine: affine transformation. Dropout: random dropping of units including connections.)

corresponds to the x and y components of $\{\vec{d}_1, \vec{d}_2, \ldots, \vec{d}_{50}\}$. The first layer is a convolutional layer [28] with 50 filters with a size of 3 (holes) × 5 (rows) × 2 (channels) that is connected to the second layer with 450 units. The last part of each NN comprises the third layer (200 units), the fourth layer (50 units), and the output layer (one unit). These layers are fully connected through rectified linear units (ReLU [29] and affine transformations. Stochastic information selection units (DROPOUT [30]) are additionally inserted between the third and fourth layers. The single output unit is intended to predict $\log_{10}(Q(\vec{x}))$.

3.2 Learning phase

[Step (IV)]: For this phase, we employ a conventional loss function L consisting of two terms: the squared difference between the output of the NN and the teacher data (i.e. $\log_{10}(Q(\vec{x}_i))$), and the summation of the squared connection weights w_m in the network (weight decay method [31]), where the latter is used to avoid the overfitting,

$$L = \left|\text{Output}(i) - \log_{10} Q(\vec{x}_i)\right|^2 + \frac{1}{2}\lambda \sum_m w_m^2 . \quad (1)$$

For the hyperparameter λ we use 0.00333 determined from the (10-fold) cross-validation method. In the training process, we randomly select one set $\{\vec{x}_i, \log_{10} Q(\vec{x}_i)\}$ from the training dataset $\{\vec{x}_{1:j}, \log_{10} Q(\vec{x}_{1:j})\}$, where j is the number of samples in the present dataset as defined in Figure 1, and change the internal parameters of the NN to reduce L using the back-propagation method [32]. Here, the actual output of the NN is referred to as $\log_{10} Q_{\text{NN}}$, where Q_{NN} is an approximation of the Q factor. We apply the momentum optimization method to speed up convergence [33], where the learning rate and the momentum decay rate are set to 1.0×10^{-4} and 0.9, respectively. The random selection of one structure and following reduction of L by using the back-propagation method is repeated 5×10^4 times. Ten separate NNs are trained by the same method, but with different orders of data feeding. Therefore, after the training, each NN has acquired different internal parameters, which widens the divergence of the candidate structures that are generated in the following step (V).

3.3 Structure search phase

[Step (V)]: Several candidate structures (here, we use $m = 70$) with potentially higher Q factors are generated using the gradient method. For this we define the loss function L',

$$L' = |\log_{10} Q_{\text{target}} - \log_{10} Q_{\text{NN}}|^2 + (\text{Artifitial loss}), \quad (2)$$

and calculate the gradient of L' with respect to $\vec{x}$ (i.e. $\nabla_{\vec{x}} L'$) using the back-propagation method [32], where Q_{target} is set to a very high value (here, we use 1.0×10^8). Starting from a randomly generated initial structure defined in the parameter space by $\vec{x}_k^{\text{ini}}$ ($k > j$), we incrementally change the structure to reduce the loss L' (i.e. $\vec{x}_k \leftarrow \vec{x}_k + \Delta\vec{x}$, where $\Delta\vec{x}$ is a set of incremental hole displacements calculated from $\nabla_{\vec{x}} L'|_{\vec{x}_k}$ based on the momentum method [33]), which is repeated 2×10^4 times. The artificial loss or regularization term in Eq. (2) is used to constrain the structural parameter space that is explored during the optimization, and different conditions are used to obtain different candidate structures. We designed the following three types of artificial losses, where λ' is a control parameter.

(A) Squared distance from the base structure or the best structure in the previous round:

$$\frac{1}{2}\lambda' |\vec{x} - \vec{x}_b|^2, \quad (3)$$

where $\vec{x}_b$ refers to the sample structure with the highest Q in the previous rounds (i.e. the highest Q among $Q(\vec{x}_{1:j})$). (In the case of the first round, $\vec{x}^b$ is set to 0, because the base structure has no displacements). This artificial loss is designed to explore the parameter space in the vicinity of the best structure in the previous rounds.

(B) Squared distance from a randomly generated initial structure $\vec{x}_k^{\text{ini}}$:

$$\frac{1}{2}\lambda' |\vec{x} - \vec{x}_k^{\text{ini}}|^2 . \quad (4)$$

This artificial loss forces exploration of unknown parameter space stochastically. It is expected that a structure with a higher Q that is not predictable from the training dataset can be accidentally found by using this artificial loss.

(C) Sum of the inverse of the distances from all the structures in the training data set:

$$\lambda' \sum_{i \le j} |\vec{x} - \vec{x}_i|^{-1}. \quad (5)$$

This artificial loss increases as the parameter space vector of the structure that is being optimized approaches the locations of the known structures, $\vec{x}_i$ with $i \le j$. This restriction forces exploration of unknown parameter space more strictly than (B).

For the present demonstration, we designed and investigated the following three strategies of candidate generation:

Strategy (A): Each NN generates seven different candidates using the artificial loss (A) with seven different λ' (3.0, 1.0, 0.1, 0.01, 0.001, 0.0001, 0.00001).
Strategy (A+B): Each NN generates three candidates using the artificial loss (A) with three different λ' (1.0, 0.1, 0.01), one candidate without using artificial losses ($\lambda' = 0$), and three different candidates using the artificial loss (B) with three different λ' (1.0, 0.1, 0.01).
Strategy (A+C): Each NN generates three candidates using the artificial loss (A) with three different λ' (1.0, 0.1, 0.01), one candidate without using artificial losses ($\lambda' = 0$), and three different candidates using the artificial loss (C) with three different λ' (1.0, 0.1, 0.01).

3.4 Validation and update phases

[Step (VI)]: The Q factors of the 70 candidate structures obtained in step (V) for each strategy are determined by 3D-FDTD calculations. The calculation conditions are the same as in [19].

[Step (VII)]: The new data consisting of 70 candidate structures (defined by $\vec{x}_{j+1:j+70}$) and their Q factors ($Q(\vec{x}_{j+1:j+70})$) calculated in (VI) for each strategy are added to each strategy's training dataset.

[Step (VIII)]: Steps (IV)–(VII) are repeated 101 times. During this iterative optimization of the regression function $Q_{NN}(\vec{x})$ and, consequently, also that of the cavity design, different series of training datasets are accumulated for each strategy, and 8070 sample cavities are accumulated in each dataset after 101 rounds of optimization.

3.5 Results

Figure 4 shows the highest Q factors of the additional 70 sample structures generated in each iteration step

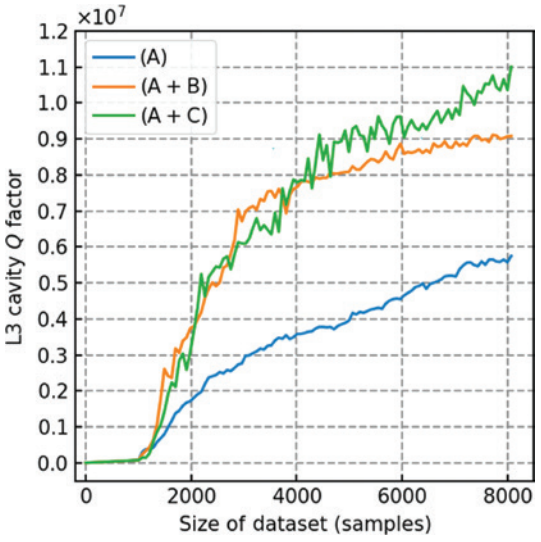

Figure 4: The highest Q factor of the additional 70 sample cavities generated in one round as a function of the size of the dataset. The results for the three different strategies (A), (A+B), and (A+C) are shown with blue, orange, and green curves, respectively. The highest Q factors of the candidates identified in 101 rounds of optimization of the regression function are 5.75×10^6, 9.12×10^6, and 1.10×10^7 for strategies (A), (A+B), and (A+C), respectively.

cycle as a function of the size of the dataset (training set + 70 structures generated in that cycle). The corresponding Q_{NN} are not discussed in the following, because the regression function is only employed to identify structures with potentially higher Q factors (via the gradient method). The results for the different strategies (A), (A+B), and (A+C) are shown with the blue, orange, and green curves, respectively. We find that the highest Q achieved in each round overall increases with further iteration, although some fluctuations exist. The highest Q factors of the structures that have been detected by 101 iterations of cavity design optimization are 5.75×10^6, 9.12×10^6, and 1.10×10^7 for strategies (A), (A+B), (A+C), respectively. These values are larger than the Q factor of the original structure (Figure 2) by factors of about 800–1500. Figure 5 plots the inter-structure distances between the best structure in the present round and the best structure in the previous rounds in terms of the parameter space vector $\vec{x}$, indicating how large the modifications in each round of optimization are. It can be confirmed that the inter-structure distances tend to decrease as the optimization proceeds. The inter-structure distances for strategy (A+C) is basically always larger than those for the other structures, and that for strategy (A+B) is larger than that for (A) only at early stages (<4000 samples). The air hole displacements of the structures with the highest Q factors found during 101 optimization cycles for the three strategies

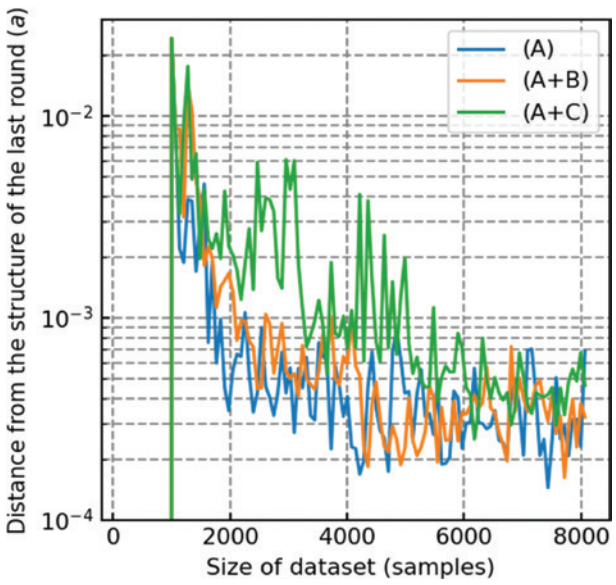

Figure 5: Inter-structure distances between the structure with the highest Q factor in the present round (at $\vec{x}_b'$ in the parameter space) and that in the previous rounds (at $\vec{x}_b$) as a function of the size of the dataset.
The inter-structure distance $|\vec{x}_b' - \vec{x}_b|$ indicates how large the modifications in each round of optimization are. The results for the three different strategies (A), (A + B), and (A + C) are shown with blue, orange, and green curves, respectively.

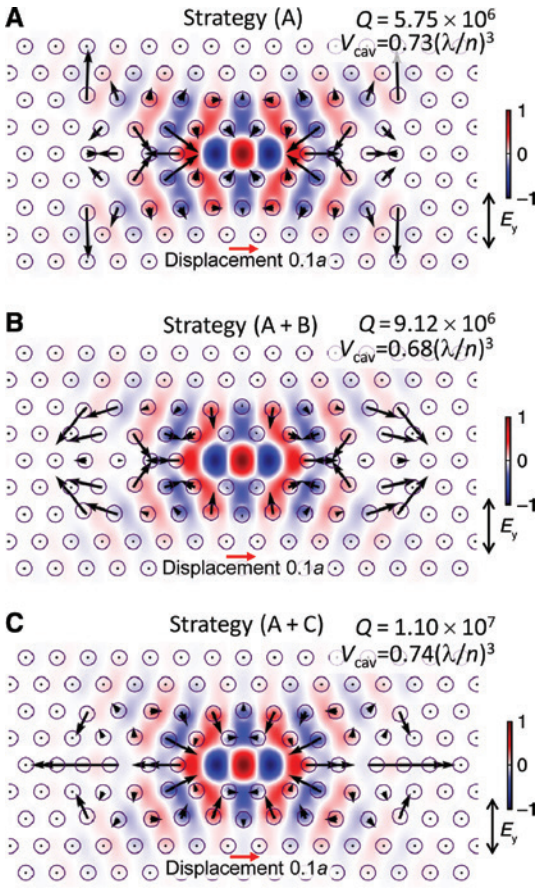

Figure 6: Air hole displacements of the optimized cavities.
The results for three different strategies (A), (A + B), and (A + C) are shown in (A), (B), and (C), respectively. The circles represent the determined optimum positions of the air holes, and the arrows represent the displacement vectors with the scale shown by the red arrow. The electric field distributions of E_y are also plotted in color. The highest Q factors of the structures generated by our proposed method during 101 iteration cycles are 5.75×10^6, 9.12×10^6, and 1.10×10^7 for strategies (A), (A + B), (A + C), respectively.

are shown in Figure 6. The distribution of E_y is shown as well. It is interesting to note that the displacements of the best cavities for the three strategies are significantly different. The modal volumes of the cavity modes are also provided in Figure 6: the V_{cav} of the optimized cavities are 0.73, 0.68, and 0.74 $(\lambda/n)^3$ for strategies (A), (A + B), (A + C), respectively, which are slightly larger than that of the base structure [0.61 $(\lambda/n)^3$ as shown in Figure 2]. We thus confirmed that the increase in V_{cav} is less than 22%, which is much smaller than the presently achieved increase in Q (increase by a factor of about 10^3). Therefore, the optimization of Q in the present case is almost equivalent to the optimization of Q/V_{cav}.

4 Discussion

4.1 Performance of the three strategies

At first, we compare the results of the iterative optimization proposed in this report with the optimization results of the previously reported one-round NN-based optimization method [19]. For accurate comparison, the same dataset sizes need to be employed. The size of the training dataset that is utilized in different iteration steps of the new optimization method exceeds 1000 after the first round. On the other hand, the number of random cavities in the training dataset that can be utilized for the verification of the previous optimization method is only 1000. Therefore, we generated 1050 random sample cavities in addition to the initially prepared 1000 random cavities (2050 cavities in total), calculated their Q factors, and performed the previously proposed optimization method [19] using these data. The following conditions were used for this optimization: only one NN is employed, and this NN generates five different candidates using the artificial loss (A) with five different λ' (1.0, 0.5, 0.1, 0.05, 0.01), and the structure with the highest Q among the 2050 cavities (No. 158, $Q = 8.12 \times 10^4$) serves as the initial structure for the

gradient method, in order to reproduce the condition used in Ref. [19]. The Q factor of the best structure obtained with this method is 1.41×10^5. This result has to be compared with the result of the 16th round of the present iterative optimization method, because here the size of the training dataset equals 2050. As shown in Figure 4, the Q factors of the best structures after the 16th round of the iterative optimization method are 1.91×10^6, 3.99×10^6, and 4.32×10^6 for the strategies (A), (A + B), and (A + C), respectively. The values obtained by the new method are 13–30 times larger than the one obtained by the previous method. This evidences that the proposed iterative method is very effective compared to the previous method.

The reason for this significantly different behavior is explained with the distributions of Q factors in the training datasets provided in Figure 7. In the case of random structure generation, the distribution of the Q factors in the dataset does not change largely even if we increase the number of the random structures from 1000 (Figure 7A) to 2050 (Figure 7B). Because the Q factors of cavities optimized by extrapolation cannot largely exceed those in the dataset, it is difficult to increase the Q factor by simply increasing the number of randomly generated sample structures. In the case of the new method we employed the distribution in Figure 7A as initial training set, but in the course of the 15 iteration cycles (resulting in 1050 new structures generated by the NNs), the distribution of the Q factors at the learning phase of the 16th round significantly expands to higher values as shown in Figure 7C. This result evidences that iterative expansion of the dataset is more effective than to simply increase the number or randomly generated structures.

The effect of iterative optimization is more apparent when we compare the learning result for the initial dataset and that for the 101st round of the optimization. Figure 8A shows an example of the correlation between Q_{NN} (predicted by the NN) and the actual Q (calculated by FDTD) after learning the initial 1000 random structures. A good correlation between Q_{NN} and Q can be confirmed. Therefore, the regression functions obtained by training NNs are useful for extrapolation of new candidate structures [19]. However, only structures with Q factors less than 8×10^4 are included in the initial dataset. The color scale indicates the frequency of samples structures within a certain range of Q factors: the yellow region in Figure 8A indicates that most sample structures have a Q slightly less than 10^4. Therefore, it is impossible to extrapolate structures with Q factors on the order of 10^7 from the initial dataset only. To solve this issue, we proposed the use of iterative expansion of the dataset. Figure 8B shows the correlation between Q_{NN} and Q after learning the 8000 structures that include 7000

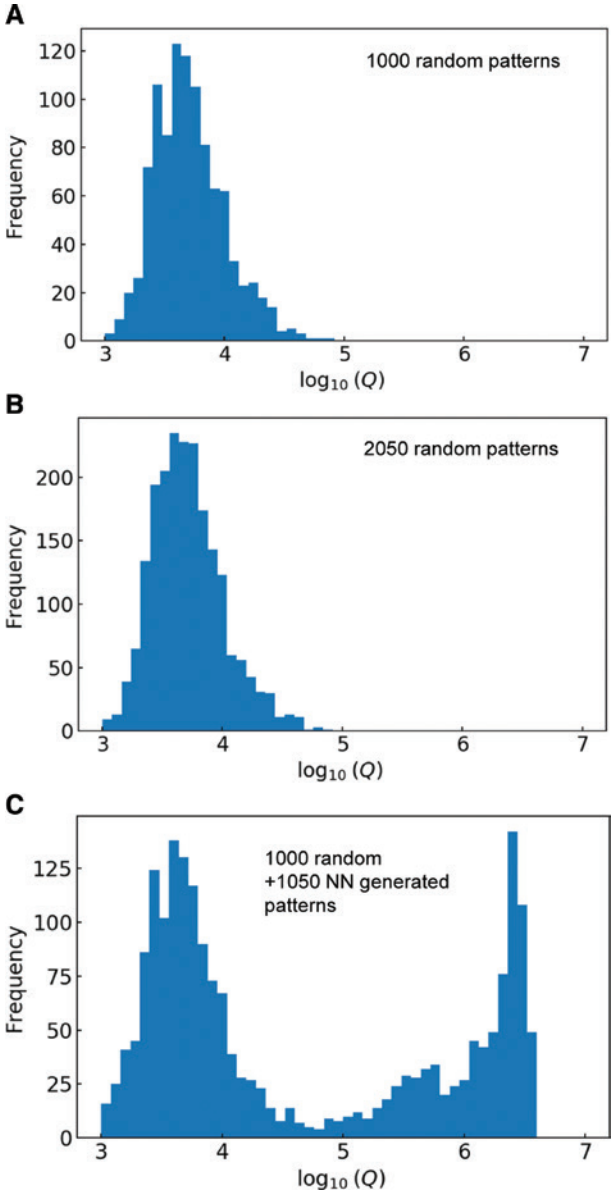

Figure 7: Histograms of the Q factors in the training datasets generated by different methods.
(A) The initial 1000 random patterns. (B) The initial 1000 pandom patterns + 1050 newly generated random patterns. (C) The initial 1000 random patterns + 1050 patterns generated by the regression function provided by the NN according to strategy (A + C).

new structures generated according to strategy (A + C). Also here, the correlation is good, and additionally a considerable amount of sample structures near $Q = 10^7$ can be confirmed. This demonstrates that the new iterative method successfully generated additional sample structures that are necessary for large improvements of Q factors.

In the following we compare the three strategies with respect to the best Q and the computational costs. The best Q factors found with strategies (A + B) and (A + C) are about

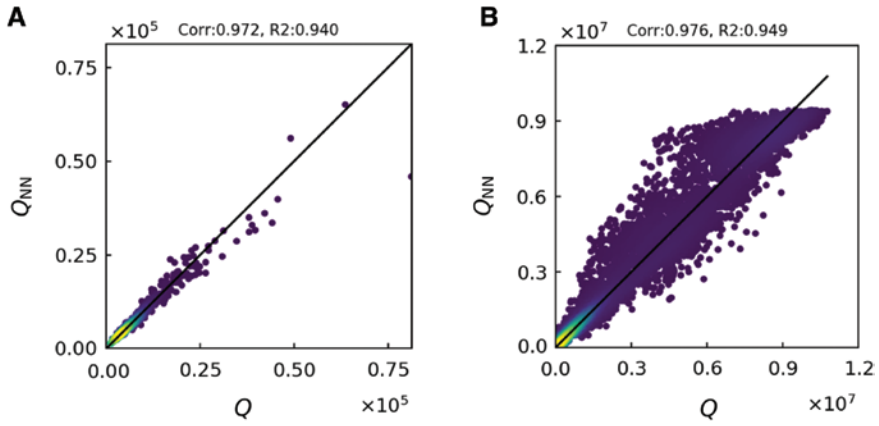

Figure 8: Correlation between Q factors calculated by the 3D-FDTD method (Q) and predicted by neural networks (Q_{NN}) at different stages. (A) The correlation after the learning phase of the first round where the initial 1000 random structures have been used as training dataset. (B) The correlation after the learning phase of the 101st iteration step of strategy (A + C) where 7000 automatically generated structures and the initial 1000 random structures have been used in the training. The correlation coefficients and the decision coefficients are provided on the upper side of the graphs.

1.6 and 1.9 times larger than that of (A). The differences in the improvement ratios originate from the differences in the methods of generating candidate cavity structures. In strategy (A), candidates for high Q structures are generated by exploring the parameter space following the gradient of Q_{NN} predicted by the trained NN while keeping the distance from the best structure in the previous rounds small. This constraint is controlled by parameter λ', which was changed from 3 to 1×10^{-5} to generate seven different candidates. Candidates with higher Q factors are frequently generated, but the candidate structures are generated according to the past experience (although some randomness is introduced by the initial structures). Therefore, the possibility to get stuck in a local maximum during the repetition of the optimizations is relatively large. The rapid decrease of the inter-structure distance for this strategy shown in Figure 5 supports this interpretation.

In strategy (A + B), half of the candidates are generated according to the past experience, and half of the candidates are generated by exploring the parameter space near randomly generated initial structures. The latter approach is expected to add diversity to the generated candidates and the training dataset. It is important to note that the latter half is not just a random generation; here, candidates are explored based on experience (the gradient of Q_{NN}), while the space to be explored is intentionally limited. This can prevent getting stuck in an already known local maximum. This explanation is supported by the larger inter-structure distances for this strategy compared to those for strategy (A) in the early stages of optimization (Figure 5; <4000 samples). It seems that the advantage of this strategy decreases as the number of iteration cycles increases as shown in Figures 4 and 5. We explain this with the higher probability of an overlap between the randomly generated initial structure and some sample in the training dataset after many iterations, reducing the diversity of the generated candidates. However, this strategy detected a structure with a Q factor that is 1.6 times larger than the best structure found with strategy (A). It is noted that the computational cost for this strategy is the same as that for (A), because the computational cost for the artificial loss (B) [Eq. (4)] equals that of the artificial loss (A) [Eq. (3)].

In strategy (A + C), half of the candidates are generated according to the past experience, and half of the candidates are generated by exploring the parameter space according to the gradient of Q_{NN} while avoiding the space near the already known structures in the training data set. Therefore, unknown parameter space is explored more explicitly compared to the case of using artificial loss (B). The maximum Q factor found with this strategy is 20% larger than that detected with strategy (A + B) as shown in Figures 4 and 6. Moreover, the tendency to detect significantly higher Q factors in the next iteration step is still not saturated even at 101 optimization cycles (Figure 4; the inclination of the green curve is relatively steep). This is in contrast to the case of (A + B) and means that strategy (A + C) can avoid local maxima more effectively. The much larger inter-structure distances of this strategy shown in Figure 4 support this interpretation. The drawback is the increase in the computational cost: as can be seen from Eq. (5), the evaluation cost of the artificial loss (C) scales with the number of samples in the training dataset (N), while the other terms in the loss function do not scale with N as can be seen in Eqs. (2)–(4). However, this evaluation has a marginal effect on the total cost for optimization

because the computational cost of the first principle calculation is much larger than that for the training and structure generation [34]. Therefore, the total computational cost for optimization can be considered proportional to the number of sample structures, which directly represents the number of first principle calculations necessary. It is also interesting to note that the evaluation cost of the artificial loss (C) scales much slower than that of the Bayesian optimization discussed later.

4.2 Comparison with other optimization methods

Here, we compare the L3 cavity optimization performances of our proposed method and other state-of-the-art optimization methods as a benchmark. The genetic algorithm based method was used in [16] to optimize five parameters in the Si-based L3 cavity and enabled detection of a structure with a Q of 4.2 million by using ~8000 sample cavities. Reference [17] optimized nine parameters in L3 cavity using a leaky component visualization method and found a structure with a Q of 5.3 million by using 200 sample cavities. In comparison, our proposed method was used to optimize 25 parameters of the L3 cavity, and we detected a structure with a Q of 11.0 million by using 8070 sample cavities generated with strategy (A + C). The maximum Q detected by the proposed method is more than 2.6 times larger than that found in [16], while the number of sample cavities that have been used is almost the same.

Compared to the leaky component visualization method, the Q obtained in the present method is more than two times larger. Although the number of sample cavities used in [17] is only about 200, these sample structures had to be explored manually, which usually consumes similar time and more effort compared to the proposed automated method. Therefore, our proposed method has provided a structure that is more optimized than those of the two previous methods, while the computational costs are similar. We consider that the higher optimization efficiency of the proposed method has two origins: a training database that contains all experiences accumulated during the whole calculation and also the aggressive search of unknown parameter space, which results in generation of candidate structures that are useful for the optimization.

From a generic point of view, other approaches based on a combination of direct and inverse NNs can be considered as well [35–40]. These methods are able to derive a target structure with given (desired) properties in case that the target response of the physical system (the input structure) can be well interpolated by a direct NN using the training dataset. However, we believe that these approaches are not very efficient in our case, because our purpose is to obtain structures with Q factors much larger than those in the initial training dataset. Therefore, it is necessary to implement a method that iteratively generates physically more significant candidate structures that are necessary for a large improvement of Q. Theoretically, it should be possible to employ NNs for the generation of such candidate structures, but prior to that we have to study suited NN learning strategies. This is considered to be an important topic of future research.

We also note that the idea of dimensionality reduction may help in some cases. For example, it has been shown that the problem of optimizing a grating structure with five tuning parameters can be reduced to search of the optimum on a 2D hyperplane [41]. This has been achieved by using an initial simple optimization via search from random starting points and sub-space identification based on the principal component analysis. The dimensionality reduction enables full search over the reduced parameter space, which requires less computational resources than the exhaustive search in the original design space. However, it is likely that this strategy is difficult to apply when the dimension of the tuning parameter is much larger (e.g. 25 dimensions as in the present case), because the characteristics of such a problem usually cannot be represented by a much smaller number of parameters.

4.3 Comparison with Bayesian optimization

Finally, we compare the proposed method and the well-known Bayesian optimization in the context of generic optimization methods. The Bayesian optimization is a powerful tool to optimize a black-box function that is expensive to calculate [42, 43]. In this method, an approximate function (usually a so-called Gaussian process [43]) that predicts not only the mean but also variance (uncertainty) of the values of the black-box function is generated by using the present dataset (the training dataset in our case). Then, an acquisition function that evaluates the probability of obtaining better values is prepared based on the predicted mean and uncertainty. As a new observation point (= candidate), the point with the highest value of the acquisition function is searched in its parameter space, and the value of the black-box function at this observation point is calculated and added to the dataset. This procedure is iterated many times. We note that the approach of our proposed method constitutes almost the same procedure. As

explained in Section 2, our framework uses a NN to construct an approximation function of the black-box function $Q(\vec{x})$. The artificial loss (C) roughly evaluates the inverse of the uncertainty, and the use of several NNs trained with different data feeding orders also corresponds to the evaluation of the uncertainty of the predicted values.

However, there are important differences between the Bayesian optimization and our proposed optimization method. One is the computational cost for the search of candidate structures in the high-dimensional parameter space (this applies to the so-called normal Bayesian optimization only), and the other is the computational cost of the learning of a large-scale training dataset (this also applies to other types of Bayesian optimization). Concerning the former difference, the normal Bayesian optimization usually uses direct search methods without relying on derivatives [43], and therefore sufficient search in a high-dimensional parameter space is impossible from the viewpoint of computational cost (a practical parameter space is usually limited to less than 10 dimensions) [43–45]. The gradient method starting from random initial points would be useful for the search in high-dimensional space, but it is difficult to implement because the gradient of the acquisition function in the normal Bayesian optimization tends to be 0 over a wider region in the parameter space as its dimension increases, because of the characteristics of the kernel function [44]. The Bayesian optimization with the elastic Gaussian process [44] can overcome this issue, but computation costs for the learning of large-scale datasets remain high as discussed later. The random embedding method [45] can also treat high-dimensional parameter spaces but only under the rather restrictive assumption that the numbers of important dimensions are very small. In contrast, our method is able to effectively utilize the gradient method in high-dimensional space because the gradient of the loss function does not disappear owing to the ReLU nonlinear layers in the NNs and the properly designed artificial loss terms. Therefore, a parallel trial of gradient-based searches starting from many randomly generated initial points works well in our case.

Regarding the relationship between the scale of the training dataset (N) and the computational cost for the training, the cost in the Bayesian optimization scales with N^3 because the inverse of a $N \times N$ matrix has to be calculated for the training [43]. (Ref. [46] utilized a deep neural network with a Bayesian linear regressor in the last hidden layer to resolve this issue. However, the maximum feasible dimension of the parameter space is still limited because direct (parallel) search is utilized [46]). Fortunately, the training cost of a NN only scales with N^1, and thus a large-scale dataset can be employed to increase the precision of the candidate search, which is especially important for the optimization in a high-dimensional parameter space. In addition, our method employs the gradient method starting from random initial points, which enables efficient exploration of a high-dimensional space. In total, it is considered that the proposed approach benefits from the characteristics of a NN-based regression, which enables training of a large-scale dataset and search in a high-dimensional parameter space while introducing the policy of the Bayesian optimization (i.e. the mean and variation of the prediction are taken into account).

4.4 Influences of fabrication errors

The deviations of the actually fabricated geometry from the intended design can be classified into the average deviations of air hole radius r and slab thickness t from the corresponding design values and the random fluctuations of radii and positions of the air holes.

Figure 9 visualizes the influences of variations in the average r and t on the Q factors of the optimized structures

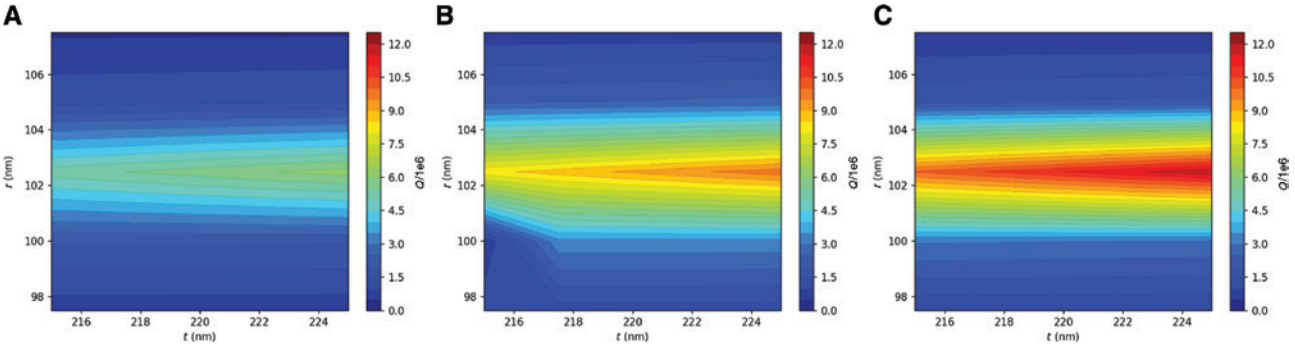

Figure 9: The influences of changes in the air hole radii r and the slab thickness t on the Q factors.
The results for the cavity designs optimized by the strategies A, A + B, and A + C, are shown in (A), (B) and (C), respectively. The values employed in Section 3 are $r = 102.5$ nm and $t = 220$ nm.

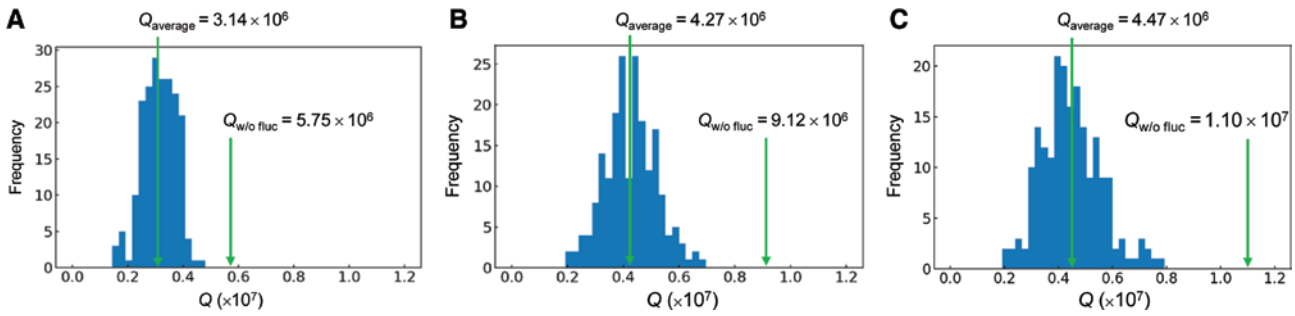

Figure 10: The influences of random fluctuations of air hole radii and positions on the Q factors.
The results for the cavity designs optimized by the strategies A, A + B, and A + C are shown in (A), (B), and (C), respectively. The random deviations follow a Gaussian distribution with a standard deviation of $a/1000$, and the same 200 random deviation patterns are applied to the cavities optimized by the three different strategies.

obtained after 101 iteration steps by the three different strategies. It can be confirmed that the Q factor depends weakly on the slab thickness, while a change of the air hole radii on the order of 2 nm is detrimental (the Q factor decreases by about 50%). Fortunately, r (and t) can be adjusted in steps on the order of about 0.5 nm (1 nm) using a post process consisting of surface oxidization and oxide removal [47].

Figure 10 shows the influence of random fluctuations of hole radii and positions on the Q factors. To obtain a reliable statistical statement, we considered 200 samples with the same design and randomly applied small changes to all air hole positions (in x and y direction) and their radii without keeping the symmetry of the structure to reflect random fluctuations due to finite fabrication accuracy. The applied random deviations follow a Gaussian distribution with a standard deviation of $a/1000$, and the same 200 random deviation patterns are applied to the cavities optimized by the three different strategies. This magnitude of the randomness corresponds to actual fabrication errors determined from experiments on state-of-the-art photonic crystal nanocavities [9]. Figure 10C clarifies that the random structural fluctuations lead to an average Q factor of 4.47×10^6 for the case of the cavity with a predicted Q factor of 1.10×10^7 in the ideal case (optimization result of Strategy A + C). For the case of the cavity with a Q factor of 5.75×10^6 in the ideal case (Strategy A), the average Q factor decreases to 3.14×10^6 due to the same structural fluctuations. Although the Q factors obviously degrade due to the random fluctuations, the average and the highest Q factors including effects of fabrication errors are still larger for cavity designs with a larger Q factor in the ideal case. Therefore, the designs with higher Q factors determined by our method have a significant impact even when fabrication errors are taken into account.

5 Conclusion

We proposed and demonstrated a new approach for optimizing 2D-PC nanocavity designs, which have large degrees of structural freedom. This approach comprises the repetition of the following four steps: training of NNs to learn the relationship between cavity structure and the Q factor using the present dataset, generation of candidate structures using the trained NNs, calculation of their Q factors, and finally adding the new structures and Q factors to the dataset. The key point of this approach is to generate a variety of candidate structures to avoid getting stuck in a local maximum in the high-dimensional parameter space. For this purpose, we prepared several NNs and trained them with different data feeding orders. In addition, we designed three artificial loss terms and used them to generate candidate structures by employing the regression function provided by a trained NN. It was demonstrated that the artificial loss term that increases near the known structures in the dataset works most efficiently to increase the speed of generating structures with higher Q factors: this method generated an optimized Si-based L3 nanocavity structure with a Q factor of 11 million (here, 25 parameters were fine-tuned using 101 iterations and a total of 8070 sample cavities). This Q factor is more than 2 times larger than the Q factors obtained by previously reported methods, while computational costs and efforts are similar. We also compared our method and the Bayesian optimization in the context of generic optimization methods. The proposed approach is effective not only for the optimization of 2D-PC nanocavity designs but also for generic optimization problems in high-dimensional parameter space. Further development of the present idea including approaches like combination with the adjoint calculation of the gradient [20] may be advantageous for many optimization problems.

Funding: Funding was provided by JSPS, funder Id: http://dx.doi.org/10.13039/501100001691, KAKENHI (19H02629), and by a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

Acknowledgments: The authors would like to thank Mr. Koki Saito for his helpful textbook on deep learning written in Japanese (Deep learning from scratch, O'Reilly Japan).

References

[1] Noda S, Chutinan A, Imada M. Trapping and emission of photons by a single defect in a photonic bandgap structure. Nature 2000;407:608–10.
[2] Akahane Y, Asano T, Song BS, Noda S. High-Q photonic nanocavity in a two-dimensional photonic crystal. Nature 2003;425:944–7.
[3] Song BS, Noda S, Asano T, Akahane Y. Ultra-high-Q photonic double-heterostructure nanocavity. Nat Mater 2005;4:207–10.
[4] Asano T, Song BS, Noda S. Analysis of the experimental Q factors (~1 million) of photonic crystal nanocavities. Opt Express 2006;14:1996–2002.
[5] Kuramochi E, Notomi M, Mitsugi S, Shinya A, Tanabe T, Watanabe T. Ultrahigh-Q photonic crystal nanocavities realized by the local width modulation of a line defect. Appl Phys Lett 2006;88:041112.
[6] Takahashi Y, Hagino H, Tanaka Y, Song BS, Asano T, Noda S. High-Q nanocavity with a 2-ns photon lifetime. Opt Express 2007;15:17206–13.
[7] Kuramochi E, Taniyama H, Tanabe T, Shinya A, Notomi M. Ultrahigh-Q two-dimensional photonic crystal slab nanocavities in very thin barriers. Appl Phys Lett 2008;93:111112.
[8] Han Z, Checoury X, Néel D, David S, El Kurdi M, Boucaud P. Optimized design for 2×10^6 ultra-high Q silicon photonic crystal cavities. Opt Commun 2010;283:4387–91.
[9] Sekoguchi H, Takahashi Y, Asano T, Noda S. Photonic crystal nanocavity with a *Q*-factor of ~9 million. Opt Express 2014;22:916–24.
[10] Asano T, Ochi Y, Takahashi Y, Kishimoto K, Noda S. Photonic crystal nanocavity with a Q factor exceeding eleven million. Opt Express 2017;25:1769–77.
[11] Asano T, Noda S. Photonic crystal devices in silicon photonics. Proc IEEE 2018;106:1–13.
[12] Srinivasan K, Painter O. Momentum space design of high-Q photonic crystal optical cavities. Opt Express 2002;10:670–84.
[13] Englund D, Fushman I, Vučković J. General recipe for designing photonic crystal cavities. Opt Express 2005;13:5961–75.
[14] Tanaka Y, Asano T, Noda S. Design of photonic crystal nanocavity with *Q*-factor of ~10^9. J Light Technol 2008;26:1532–9.
[15] Lai Y, Pirotta S, Urbinati G, et al. Genetically designed L3 photonic crystal nanocavities with measured quality factor exceeding one million. Appl Phys Lett 2014;104:241101.
[16] Minkov M, Savona V. Automated optimization of photonic crystal slab cavities. Sci Rep 2015;4:5124.
[17] Nakamura T, Takahashi Y, Tanaka Y, Asano T, Noda S. Improvement in the quality factors for photonic crystal nanocavities via visualization of the leaky components. Opt Express 2016;24:9541–9.
[18] Minkov M, Savona V, Gerace D. Photonic crystal slab cavity simultaneously optimized for ultra-high Q/V and vertical radiation coupling. Appl Phys Lett 2017;111:131104.
[19] Asano T, Noda S. Optimization of photonic crystal nanocavities based on deep learning. Opt Express 2018;26:32704–16.
[20] Molesky S, Lin Z, Piggott AY, Jin W, Vucković J, Rodriguez AW. Inverse design in nanophotonics. Nat Photonics 2018;12:659–70.
[21] Lu J, Vuckovic J. Inverse design of nanophotonic structures using complementary convex optimization. Opt Express 2010;18:3793–804.
[22] Takezawa A, Kitamura M. Cross-sectional shape optimization of whispering-gallery ring resonators. J Lightwave Technol 2012;30:2776–82.
[23] Lin Z, Liang X, Lončar M, Johnson SG, Rodriguez AW. Cavity-enhanced second-harmonic generation via nonlinear-overlap optimization. Optica 2016;3:233–8.
[24] Lin Z, Lončar M, Rodriguez AW. Topology optimization of multi-track ring resonators and 2D microcavities for nonlinear frequency conversion. Opt Lett 2017;42:2818–21.
[25] Frei WR, Johnson HT, Choquette KD. Optimization of a single defect photonic crystal laser cavity. J Appl Phys 2008;103:033102.
[26] Liang X, Johnson SG. Formulation for scalable optimization of microcavities via the frequency-averaged local density of states. Opt Express 2013;21:30812–41.
[27] Maeno K, Takahashi Y, Nakamura T, Asano T, Noda S. Analysis of high-Q photonic crystal L3 nanocavities designed by visualization of the leaky components. Opt Express 2017;25:367–76.
[28] LeCun Y, Boser B, Denker JS, et al. Handwritten digit recognition with a back-propagation networks. In: Proceedings of Advances in Neural Information Processing Systems, Denver, CO, USA, 1990:396–404.
[29] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 2011:315–23.
[30] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15:1929–58.
[31] Krogh A, Hertz JA. A simple weight decay can improve generalization. In: Proceedings of Advances in Neural Information Processing Systems, Denver, CO, USA, 1991:950–7.
[32] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323:533–6.
[33] Polyak BT. Some methods of speeding up the convergence of iteration methods. USSR Comput Math Math Phys 1964;4:791–803.
[34] For the first principles simulation phase we use 70 runs of the FDTD simulation of the 3D electric and magnetic fields (12 components) in the new 70 cavity structures. These runs can be processed in parallel depending on the computation resources. Each FDTD simulation comprises 80,000 steps of spatial derivative calculations for the PC nanocavity described by a mesh size of $480 \times 320 \times 170$ cells. The training phase employs 10 runs (which can be processed in parallel), each comprising 50,000 steps of the forward and backward

calculations of the multiplications of vectors and matrices with sizes of {vector, matrix}={50, 50×50}×9 (convolution), {450, 450×200}, {200, 200×50}, and {50, 50×1}. The structure generation phase uses 70 runs (which can be processed in parallel), each comprising the 20,000 steps of the same vector-matrix calculations to obtain the gradient of Q of each structure during candidate generation. The loss factor of type C involves an additional cost of calculating the distances between the present structure and all the structures in the training dataset, where the dimension of each structure vector is 100 and the number of structures in the training dataset changes from 1000 to 8000 in this work. This additional cost can be on the same order as the cost for the calculation of the gradient of Q. In total, the computational cost of the first principles calculations is larger than that for the learning and structure generation phases by a factor of roughly 100.

[35] Pilozzi L, Farrelly FA, Marcucci G, Conti C. Machine learning inverse problem for topological photonics. Commun Phys 2018;1:57.

[36] Peurifoy J, Shen Y, Jing L, et al. Nanophotonic particle simulation and inverse design using artificial neural networks. Sci Adv 2018;4:eaar4206.

[37] So S, Rho JM. Simultaneous inverse design of materials and structures via deep learning: demonstration of dipole resonance engineering using core-shell nanoparticles. ACS Appl Mater Interfaces 2019;11:24264–8.

[38] Liu D, Tan Y, Khoram E, Yu Z. Training deep neural networks for the inverse design of nanophotonic structures. ACS Photonics 2018;5:1365–9.

[39] Liu Z, Zhu D, Rodrigues SP, Lee KT, Cai W. Generative model for the inverse design of metasurfaces. Nano Lett 2018;18:6570–6.

[40] Long Y, Ren J, Li Y, Chen H. Inverse design of photonic topological state via machine learning. Appl Phys Lett 2019;114:181105.

[41] Melati D, Grinberg Y, Dezfouli MK, et al. Mapping the global design space of nanophotonic components using machine learning pattern recognition. Nat Commun 2019;10:4775.

[42] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Global Optim 1998;13:455–92.

[43] Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N. Taking the human out of the loop: a review of Bayesian optimization. In: Proc. IEEE 104, 2016:148–75.

[44] Rana S, Li C, Gupta S, Nguyen V, Venkatesh S. High dimensional bayesian optimization with elastic gaussian process. In: Proc. of the 34th Int. Conf. on Mach. Learn. 70, 2017:2883–91.

[45] Wang Z, Hutter F, Zoghi M, Matheson D, De Freitas N. Bayesian optimization in a billion dimensions via random embeddings. J Artif Intell Res 2016;55:361–7.

[46] Snoek J, Rippel O, Swersky K, et al. Scalable Bayesian optimization using deep neural networks. In: Proceedings of the 32nd International Conference on Machine Learning, PMLR 37, Lille, France, 2015:2171–80.

[47] In Ref. [10], we reduced the resonant wavelength of the cavities in steps of about 3 nm by using surface oxidization and oxide removal. Theoretical calculations revealed that this wavelength change corresponds to a change in the radii of the air holes by about 0.5 nm (and a simultaneous change in the slab thickness by about 1 nm).