SCIENCE CHINA
Information Sciences

June 2023, Vol. 66 160410:1–160410:13
https://doi.org/10.1007/s11432-022-3757-x

• RESEARCH PAPER •
Special Topic: Two-Dimensional Materials and Device Applications

# AtomGAN: unsupervised deep learning for fast and accurate defect detection of 2D materials at the atomic scale

Danpeng CHENG[1], Wuxin SHA[2], Zuo XU[3], Shide LI[3], Zhigao YIN[3], Yuling LANG[3], Shun TANG[1] & Yuan-Cheng CAO[1*]

[1]*State Key Laboratory of Advanced Electromagnetic Engineering and Technology, School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan* 430074*, China;*
[2]*School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan* 430074*, China;*
[3]*CITIC Dicastal Co., Ltd., Qinhuangdao* 066011*, China*

Received 16 December 2022/Revised 28 February 2023/Accepted 5 May 2023/Published online 17 May 2023

**Abstract** The extraction of atomic-level material features from electron microscope images is crucial for studying structure-property relationships and discovering new materials. However, traditional electron microscope analyses rely on time-consuming and complex human operations; thus, they are only applicable to images with a small number of atoms. In addition, the analysis results vary due to observers' individual deviations. Although efforts to introduce automated methods have been performed previously, many of these methods lack sufficient labeled data or require various conditions in the detection process that can only be applied to the target material. Thus, in this study, we developed AtomGAN, which is a robust, unsupervised learning method, that segments defects in classical 2D material systems and the heterostructures of $MoS_2/WS_2$ automatically. To solve the data scarcity problem, the proposed model is trained on unpaired simulated data that contain point and line defects for $MoS_2/WS_2$. The proposed AtomGAN was evaluated on both simulated and real electron microscope images. The results demonstrate that the segmented point defects and line defects are presented perfectly in the resulting figures, with a measurement precision of 96.9%. In addition, the cycled structure of AtomGAN can quickly generate a large number of simulated electron microscope images.

**Keywords** deep learning, generative adversarial network, defect detection, atomic resolution, 2D materials

**Citation** Cheng D P, Sha W X, Xu Z, et al. AtomGAN: unsupervised deep learning for fast and accurate defect detection of 2D materials at the atomic scale. Sci China Inf Sci, 2023, 66(6): 160410, https://doi.org/10.1007/s11432-022-3757-x

## 1 Introduction

Defects in crystal materials destroy the crystalline periodic structure, thereby regulating various physical and chemical properties of materials that may be used to manufacture devices with diverse functions [1,2]. Point defects [3,4] of different types, locations, and densities, and line defects [5,6] with different structures can change the electronic tunability and mechanical flexibility of materials. If the density of different defects could be evaluated, it would be possible to develop an optimal manufacturing technique to improve semiconductor devices [7,8]. Thus, atom-scale defect analysis is crucial for both scientific research and industrial applications. Due to the development of transmission electron microscopy (TEM), researchers can observe such point and line defects at the atomic-level and then analyze the samples. Traditionally, TEM images have been analyzed manually by electron microscopy experts based on prior knowledge of different crystal structure characteristics [9]. However, this method is only effective for a small number of images with a limited number of atoms because the results are not credible for individual bias and the process is slow. In addition, the lab and imaging processing environments, the selection of electron

* Corresponding author (email: yccao@hust.edu.cn)

 

microscopy parameters, and slight errors in sample preparation can blur images. Low-quality images hinder defect localization and the corresponding analysis of defects.

Current efforts to apply machine learning to electron microscope images have realized automatic detection of defect positions, thereby helping investigators conduct both qualitative and quantitative defect analyses. Previous studies have used a supervised learning algorithm [10, 11] that considered defect detection as a pixel-by-pixel object detection task. However, these methods are narrowed to scarce tagged training datasets because experts must identify the defect location and type accurately when marking data [10], which is a time-consuming process. In addition, the same number of different types of defect images are required. Thus, researchers have turned their attention to unsupervised learning techniques. Unsupervised learning models may have a more far-reaching impact because they can discover the main features inherent in the material proactively; thus, the relationship between properties and structures can be identified. A set of unsupervised algorithms depends on the perfect unit cell structure. Authors assume that the majority of a TEM image belongs to a single class, i.e., the perfect unit cell, and those that differ from this class are considered defects. This process is also known as an anomaly detection problem. Commonly, the models used for anomaly detection belong to binary classifiers. For example, the one-class support vector machine utilizes a kernel function to circle the perfect unit cell inside a hyperplane, and the parts outside this circle are considered defects [12]. Another method trains the model using only defect-free images in an attempt to analyze the true distribution of defect-free data [13]. However, both of these methods cannot detect line defects when multiple phases are presented in the image.

In contrast, a generative adversarial network (GAN) [14] is suitable for multi-phase defect detection. The GAN can learn a mapping between visually distinct classes, e.g., TEM images and segmented multi-phase results. The GAN comprises a generator and a discriminator, where the generator generates instances similar to the original data, and the discriminator distinguishes the generated instance from the real image. In the basic concept of an adversary, both models are optimized alternately to improve their capabilities, ultimately making the discriminator unable to distinguish the difference between the generated and real images.

GANs have been widely used in industrial fields, e.g., small dataset augmentation [15, 16] and material surface defect detection [17–19]. In this paper, we propose a novel GAN for atom image analysis, which we refer to as AtomGAN. The proposed AtomGAN can segment line and point defects simultaneously, as shown in Figure 1. Considering the rarity of real TEM images, the model is trained using images simulated by Material Studio and the improved abTEM package [20]. Here, class one is the simulated TEM images, and class two is the images of the segmented point and line defects. The model distinguishes different types of defects by different degrees of brightness, thereby achieving the effect of classification. In addition, different classes of images do not need to be paired because the model evaluates the original image and the generated image by adding the loss of cyclic consistency from CycleGAN [21] to build a mirrored structure rather than the original image and the marked pixel image. A class encoder is also implemented to extract the features of the target class, which helps the generator generate images that are more similar to the target class. In addition, the discriminator realizes multiclass distinction through multiple channels. The proposed model exhibits high accuracy even on very blurry images, which demonstrates the potential of using a GAN model in advanced material microstructure characterization tasks.

## 2 Methodology

### 2.1 Data acquisition

The first step in developing a model is data acquisition. Due to the difficulty associated with obtaining experimental data, simulated TEM photos were used in this study to train and verify our model. Then, the model was tested using real images. Note that there are two modes of TEM imaging, i.e., parallel illumination (traditional TEM with a high-resolution TEM) and focused probe (scanning TEM (STEM)). Fortunately, for atomic resolution imaging, our defect detection algorithm applies to both modes because the model learns the atomic arrangement rules with and without defects, which is unrelated to the imaging method. Thus, we only take the STEM mode as an example to construct a dataset of electron microscope images with and without defects for network learning.

First, Materials Studio, a material calculation software produced by Accelrys in the United States, was

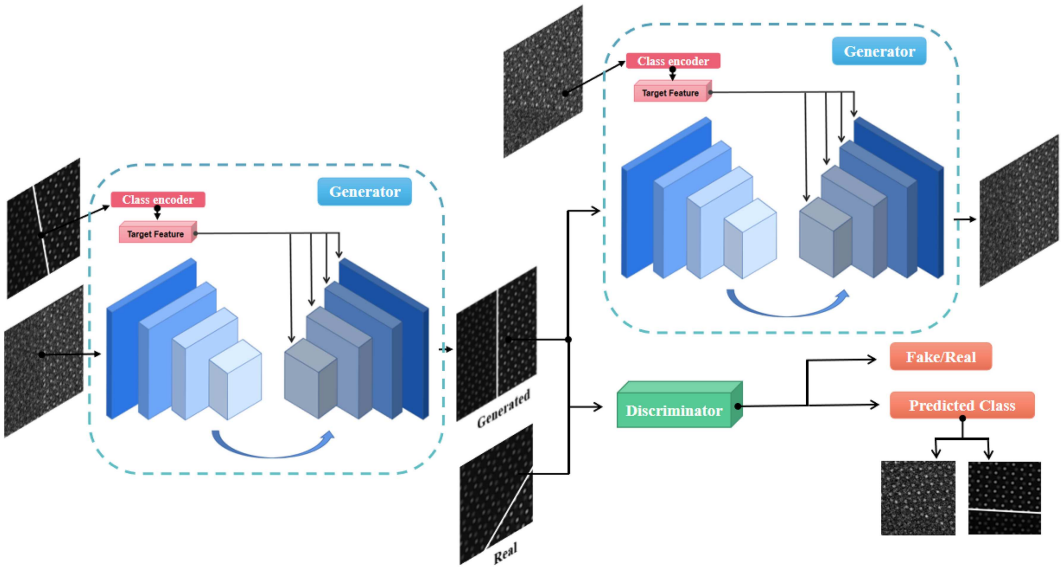


**Figure 1** (Color online) Architecture of the proposed AtomGAN for atomic scale defect detection.

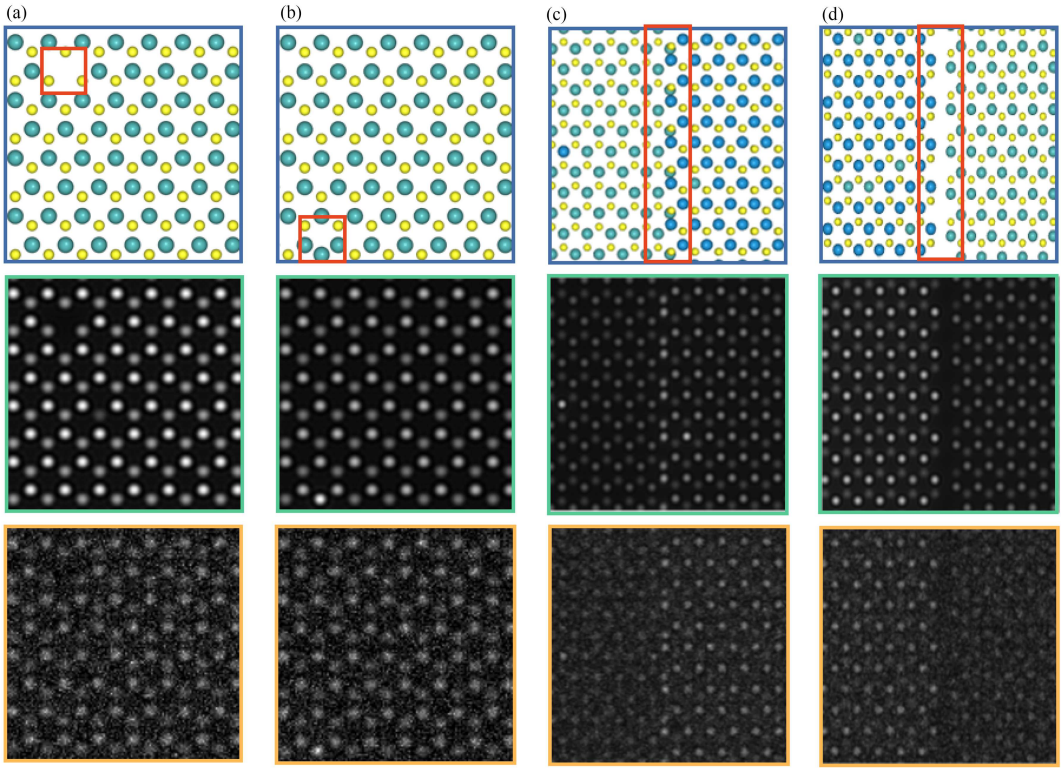


**Figure 2** (Color online) Simulated datasets. (a) and (b) are point defects; (c) and (d) are line defects.

used to build atomic models, including normal and defective structures. Point defects were created by randomly deleting atoms on the structure (Figures 2(a) and (b)). For the line defects, e.g., heterojunction, the average value of the cell parameters of two materials was selected for the new lattice parameters using the function of build layers. Then, a vacuum layer was added to the system, and highly symmetrical initial structures, e.g., hollows and bridges, were established by translating and rotating the relative positions between the two materials, i.e., $WS_2$ and $MoS_2$, as shown in Figures 2(c) and (d). The final structure is then imported as an atom object in a structure file.

Second, we employed the improved abTEM package to achieve simulated imaging. Here, the wave function was selected as a plane wave and transferred to the image plane via a contrast transfer function

**Table 1** Number of defective images and defects in both training and test datasets

| Defect type | Training | | Test | |
|---|---|---|---|---|
| | Number of defective images | Number of defects | Number of defective images | Number of defects |
| Point defect | 3984 | 6610 | 1330 | 3380 |
| Line defect | 3984 | 3984 | 1330 | 1330 |

(CTF), where defocus corresponded to a reasonable number of parameters, including the chromatic aberration parameters, residual spherical aberration, and focal spread. Finally, the experimental signal was simulated using Gaussian blur and Poisson noise. Here, the Gaussian blur was used to blur the boundaries of the atoms, and the Poisson noise simulated noise in the electron microscope images. Our improved imaging results are closer to the real electron microscope images, and the time costs are reduced compared to the original abTEM package. With the proposed method, it only takes 7 s to complete the simulation imaging for an atomic model with approximately 4500 atoms.

CTF objects were created with an electron energy of 300 KV, a spherical aberration randomly selected from 1 to 3 μm and defocus range from −4 to 4 nm, and a focal spread of 30 Å. The final image was combined using the following formula:

$$\begin{aligned} &\mathrm{GB} = \mathrm{GaussianBlur}(\mathrm{image}, \mathrm{kennelsize}),\\ &\mathrm{Img}_{\mathrm{sim}} = \mathrm{GB} + \alpha \mathrm{Poisson}(\mathrm{GB}). \end{aligned} \tag{1}$$

Here, GaussianBlur represents the Gaussian blur operation on the image of the plane wave, and Poisson represents the added Poisson noise. The kernel size for GaussianBlur was between 1 to 2, and $\alpha$, i.e., the weight for the Poisson parts, ranged from 0.4 to 0.7 in 0.01 steps. The diversity of the parameters ensures the robustness of the model. The total number of simulated images was 5314, which were split into training and validation sets as described in Table 1.

Note that the simulated images are considered as a single domain (i.e., class one) of the database, and the other domain (class two) comprised the segmented images. In the segmented images, atoms are shown as circles, point defects become bright spots, and line defects are directly represented by bright lines, as shown in Figure 2(d). The images in these two domains do not need to be paired. The proposed AtomGAN can learn the features in the two domains such that the model can detect defects and generate images that are close to the real electron microscope images very quickly.

## 2.2 Model structure

The proposed AtomGAN (Figure 1) attempts to detect various crystallographic defects and generate TEM images that are similar to real images. AtomGAN comprises the class encoder, generator, and discriminator. The hyperparameters of the entire network structure are given in Table 2. The class encoder cooperates with the generator to generate images. In addition, the class encoder translates the feature of the target domain into a class code by extracting the characteristics of the target domain and guides the image generation process to approach the target domain. The generator must generate both types of images corresponding to the two classes in datasets. The discriminator distinguishes the generated images $G(x, c)$ and images $x$ that are input to the network, and images generated by the generator attempt to fool the discriminator into thinking that the generated images are true images. These three structures are described in the following.

### 2.2.1 *Generator network*

The generator network includes two mirrored generators. A single generator ($G$) comprises four upsampling blocks and four downsampling blocks. The downsampling process compresses 256 × 256 images into a 16 × 16 feature map and increases the number of channels to 512 to extract the different features of image $x$. These feature maps are then recombined via upsampling to restore the images with the same dimension as the input. Here, image $y$ from the target domain $Y$ is also input to the generator. The image $y$ is translated into class code $c$ and injected into each upsampling block. $G$ learns the mapping from $X$ to $Y$ such that the discriminator cannot distinguish the generated image $G(x, c)$ and true image $y$. As shown in Figure 3, mirror $G$ then learns a mapping from $Y$ to $X$ with $x$'s class code $\tilde{c}$.

Note that a cycle consistency loss ($L_{\mathrm{cyc}}$) [21] is recommended to properly retain the invariant features of the input.

$$L_{\mathrm{cyc}} = \bar{E}_{x,y}[|x - G(G(x, c), \tilde{c})|]. \tag{2}$$

**Table 2** Hyperparameters of AtomGAN

| Network architecture | | |
|---|---|---|
| Operation | Layers | Output size |
| Generator | | |
| Preprocessing | Conv 1 × 1, 64 | 256 × 256 |
| Encoding | [Conv 3 × 3 64; Avgpool 2 × 2 64; Conv 3 × 3 128] | 128 × 128 |
| | [Conv 3 × 3 128; Avgpool 2 × 2 128; Conv 3 × 3 256] | 64 × 64 |
| | [Conv 3 × 3 256; Avgpool 2 × 2 256; Conv 3 × 3 512] | 32 × 32 |
| | [Conv 3 × 3 512; Avgpool 2 × 2 512; Conv 3 × 3 512] | 16 × 16 |
| Decoding | [Conv 3 × 3 512; Interpolate 2 512; Conv 3 × 3 512] | 32 × 32 |
| | [Conv 3 × 3 256; Interpolate 2 256; Conv 3 × 3 256] | 64 × 64 |
| | [Conv 3 × 3 128; Interpolate 2 128; Conv 3 × 3 128] | 128 × 128 |
| | [Conv 3 × 3 64; Interpolate 2 64; Conv 3 × 3 64] | 256 × 256 |
| Image forming | conv 1 × 1, 1 | 256 × 256 |
| Class encoder/discriminator | | |
| Preprocessing | conv 3× 3, 64 | 256 × 256 |
| Feature extraction | [Conv 3 × 3 64; Avgpool 2 × 2 64; Conv 3 × 3 128] | 128 × 128 |
| | [Conv 3 × 3 128; Avgpool 2 × 2 128; Conv 3 × 3 256] | 64 × 64 |
| | [Conv 3 × 3 256; Avgpool 2 × 2 256; Conv 3 × 3 512] | 32 × 32 |
| | [Conv 3 × 3 512; Avgpool 2 × 2 512; Conv 3 × 3 512] | 16 × 16 |
| | [Conv 3 × 3 512; Avgpool 2 × 2 512; Conv 3 × 3 512] | 8 × 8 |
| | [Conv 3 × 3 512; Avgpool 2 × 2 512; Conv 3 × 3 512] | 4 × 4 |
| | Avgpool 4 × 4, 512 | 1 × 1 |
| Classification | Linear, $N$ | $K(=2)$ |

Here, $G(\cdot)$ indicates generating new images based on the input $x$ and a target image $y$, and $\bar{E}_{x,y}$ represents sampling $x$, $y$ from the data distribution, bringing the samples into the calculation, and averaging the final calculation results. The mirror symmetry structure ensures that $G(G(x,c),\tilde{c})\approx x$; thus, we can preserve

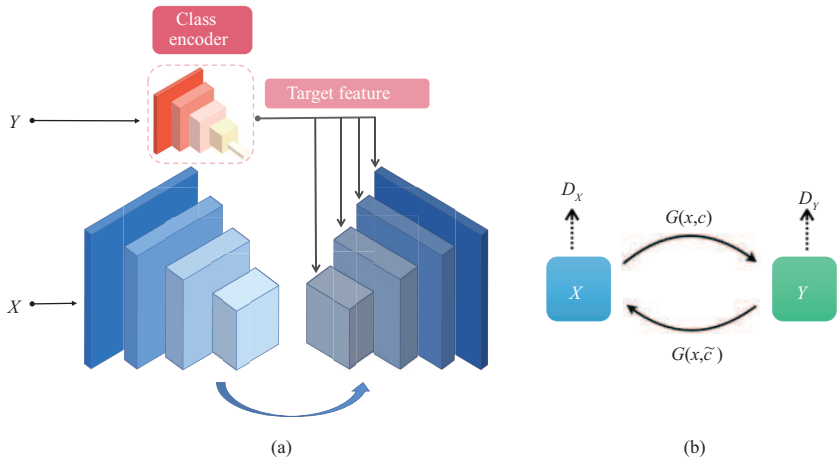


**Figure 3** (Color online) Process of the cycled generator. (a) Generator with class encoder branch; (b) mapping operations between two domains.

a large amount of information (e.g., pose information) from the original image. As a result, we only need to convert the required information without using the paired ground truth.

### 2.2.2 *Class encoder*

The code of the target domain is denoted $c$, which is obtained by a class encoder. For a given input $x$ and the corresponding target image $y$, the feature code $c = E_y(\cdot)$ is extracted from the domain $Y$. The class encoder uses a convolutional neural network (CNN) and average pooling to encode the features into a fixed size $N \times K$ vector ($N = 32$, $K = 2$), where $N$ is the dimension of the class code, and $K$ is the number of classes. To help $G$ learn effectively using the class code $c$, the class loss is introduced as follows:

$$L_c = \bar{E}_{x,y}[||c - E_y(G(x,c))||]. \tag{3}$$

Similar to the method described in [22], $E$ maps $y$ to a code $c$, which allows the generator to receive image $x$ and create a new image based on $y$. The class encoder structure helps the generated image be converted to the target domain quickly, and to some extent, solves the problem of difficult convergence in the GAN training process. In addition, the target domain can also be extended to multiple classes when $K > 2$.

### 2.2.3 *Discriminator*

The structure of the discriminator ($D$) is similar to that of the class encoder, whose output comprises $N \times K$ dimensions (Figure 4). In the last several layers of the networks, leaky ReLU functions combined with $K$ fully-connected layers are used to classify images. Here, for each class, $D$ outputs a binary result to assess whether the input is a real image or a false image of its image domain. The adversarial relationship between the generator and discriminator can be described using an adversarial loss [23] as follows:

$$L_{\text{adv}} = \bar{E}_{x,y}\left[\log D_y(x)\right] + \bar{E}_{x,y}\left[\log\left(1 - D_y\left(G\left(x,c\right)\right)\right)\right]. \tag{4}$$

Here, $D_y$ represents the output of $D$ for its class. The class encoder provides the class code $c$ for target domain $y$, and the generator uses $c$ and the input image $x$ to generate an image $G(x,c)$. Thus, the training process of $G$ maximizes the probability of $D$ making a mistake, and $D$ minimizes this processing. Therefore, the total loss $L$ can be expressed as follows:

$$L = \min_{G,E}\max_{D} L_{\text{adv}} + \lambda_1 L_{\text{cyc}} + \lambda_2 L_C. \tag{5}$$

Here, $\lambda_1$ and $\lambda_2$ are hyperparameters between 0 and 1 ($\lambda_1 = \lambda_2 = 1$ in our case), representing the weight proportion of the two losses in $L$.

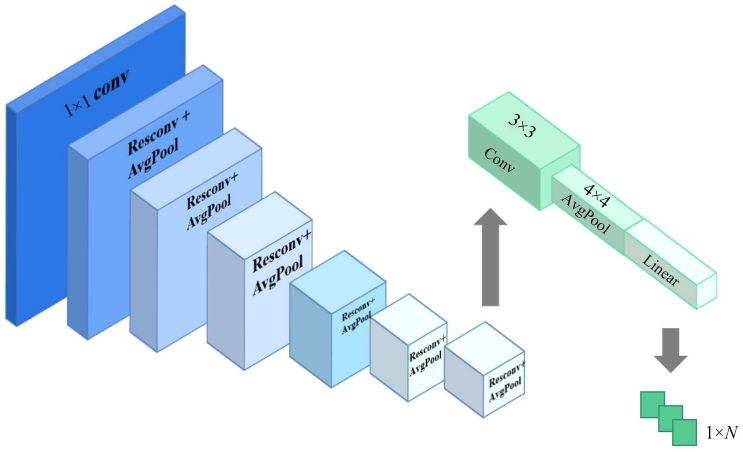


**Figure 4** (Color online) Discriminator architecture.

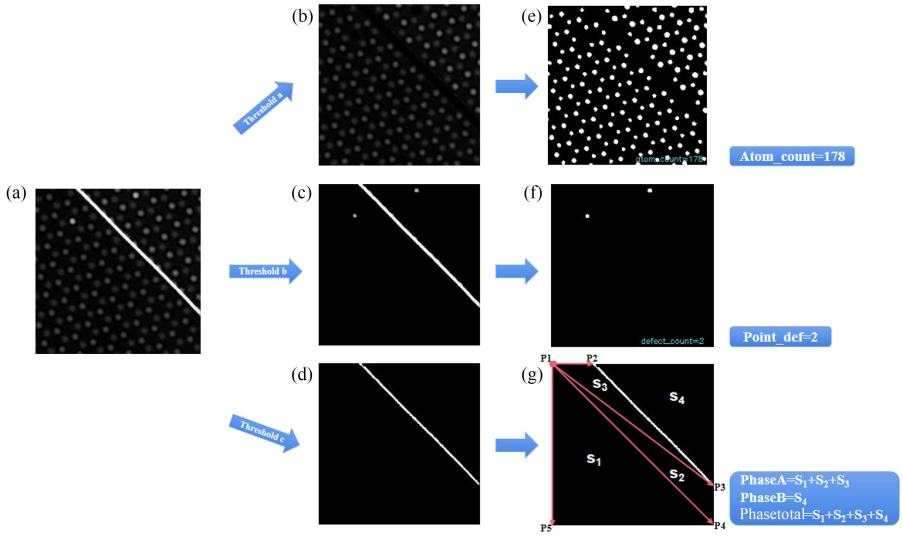


**Figure 5** (Color online) Diagram of defect visualization and quantification. (a) The output of AtomGAN; (b)–(d) the preliminary processing results using different thresholds; (e)–(g) the final segmentation results.

## 2.3 Defect visualization and quantification

The brightness values of normal atoms (Figure 5(a)), defect atoms, and boundaries differ in the segmented images; thus, they can be divided using different thresholds, as shown in Figures 5(b)–(d). For a well-segmented image, it is possible to do defect quantification and phase ratio calculations.

The images are binarized, and the atomic contours and centers of the normal and defective atoms are located separately by detecting the connected region [24]. The normal atoms, segmented point defects, and line defects are stored in different arrays. Here, a list stores all of the outline coordinates in an image. To ensure that only the shape of the atoms is preserved, we remove the list where the number of elements is less than 2 and greater than 30 (Figures 5(e) and (f)). For the number of positioned center coordinates, the proportion of the defective atoms in the image can be expressed as follows:

$$\text{Num(Defect)}/\text{Num(Total)}. \tag{6}$$

Here, Num represents both the number of defects and the number of total atoms.

In Figure 5(g), the coordinates of the grain boundaries in the marker diagram are located, and the polygon vertex coordinates are determined on each side of the boundary. Starting from the above vertices, pull out straight lines and divide the polygon into multiple triangles, so that the area of the polygon can be calculated by summing the areas of multiple triangles. Thus, the proportion of the single-phase area ($A$) can be determined as follows:

$$\text{Area}(A)/\text{Area(Total)}. \tag{7}$$

**Table 3** Specific training information for AtomGAN and CycleGAN

| Model Parameter | AtomGAN | CycleGAN |
|---|---|---|
| Main structure | Generator<br>Class encoder<br>Discriminator | Generator<br>Discriminator |
| Supported domain | More than 2 | 2 |
| Num of trainable parameters (million) | 75.3 | 11.3 |
| Generator backbone | ResBlock [27] | Resnet101 [28] |
| Normalization | Instance normalization | Instance normalization |
| Activation | LeakyReLU | LeakyReLU |
| Training w/o artificial adjustments | Without | With |
| Batch size | 4 | 4 |
| Optimizer | Adam | Adam |
| Learning rate | 1e−4 | 1e−4 |
| Training epoch | 30000 | 30000 |
| Training device | GPU | GPU |

Here, Area denotes the area calculation of the target part.

The location and density of the point and line defects play key roles in material properties. For example, the typical switching process of memristors depends on the formation or destruction of the conductive filaments in the dielectric layer caused by the change in local defect concentration [25]. The defect location and concentration calculation can better observe the switching state of semiconductor devices, help predict the direction of atomic movement, and thus analyze the relationship between semiconductor properties and defect structures [26].

## 3 Results and discussion

In this section, we present and discuss the experimental results obtained by the proposed model. First, the model's performance is evaluated from the training process and segmentation results. We found that the proposed model achieves better outcomes compared to existing algorithms. Subsequently, we utilize a confusion matrix to evaluate the defect classification accuracy. Finally, the visual effect and the advantage of the generation speed of the proposed model are described.

### 3.1 Training process description

The proposed model uses the loss of cyclical consistency from CycleGAN [21] to construct a mirrored structure. However, compared with CycleGAN, the proposed model adds unique class encoding and multichannel output. As a result, our network can realize multi-interface line defect segmentation and point defect segmentation directly, and it can solve the difficult training problem of most GANs. To ensure a fair comparison, we used the same hyperparameters for the two models, e.g., batch size, learning rate, and optimizer, as shown in Table 3 [27, 28].

Both models have trained 10000 epochs on the NVIDIA TeslaV100 SXM2, a type of graphics processing unit (GPU). Due to the introduction of the class encoder, the parameters of the proposed model multiplied compared with CycleGAN. Thus, for the generator backbone, we employed the portable ResBlock and reduced the number of layers appropriately. The results demonstrate that even if the number of generator layers is reduced, the proposed model can distinguish point defects and line defects accurately; however, the CycleGAN model cannot.

The generator training processes of the two models are visualized in Figure 6. Figure 6(a) shows an input image, and Figure 6(b) shows a target domain image. In addition, Figures 6(c)–(f) and (g)–(j) show the results generated by the CycleGAN and AtomGAN models, respectively, with 2000, 5000, 7000, and 10000 epochs. The line in these images is used to divide the two different phases, the brightest white dots indicate point defects, and different elements are represented by dots with different degrees of brightness. The corresponding average gray-scale histogram is shown below each TEM image, which displays the distribution of the pixels' values in the image. In the histograms, the horizontal axis is the gray value (brightness), where left to right represents low to high brightness. The vertical axis represents the number of pixels with the same gray value. Here, the histograms are used to visualize the similarity

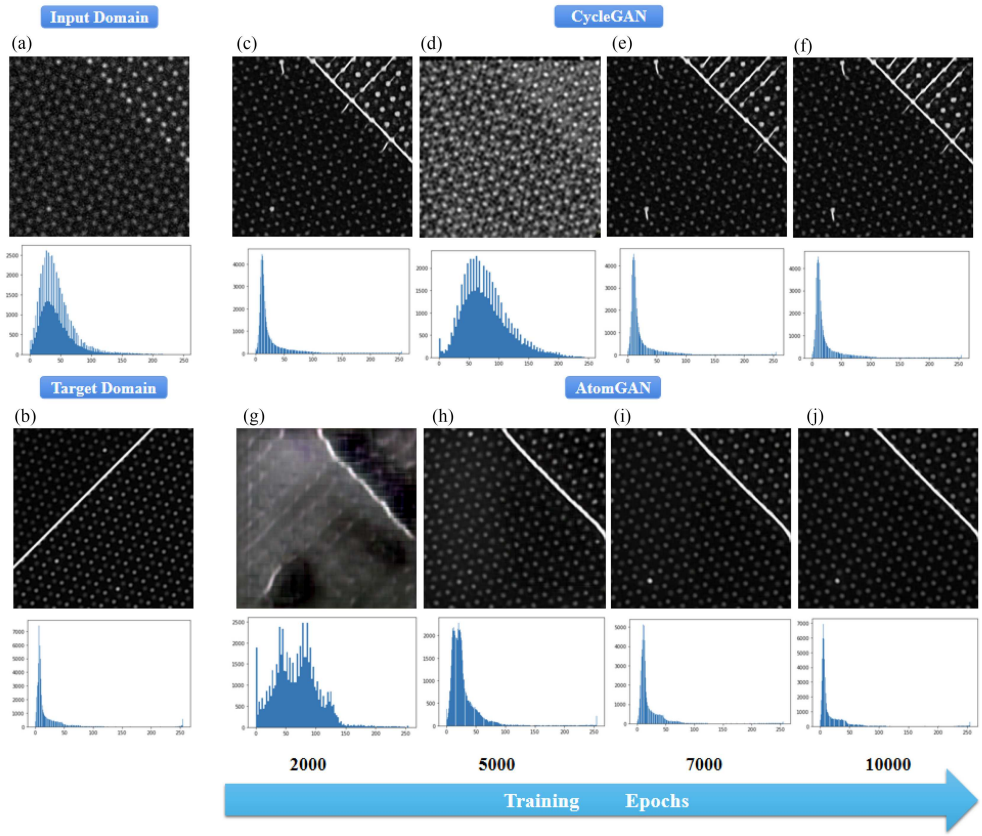


**Figure 6** (Color online) (a) The input image and (b) the target domain image correspondingly. Visualized training results of (c)–(f) CycleGAN and (g)–(j) AtomGAN obtained with 2000, 5000, 7000, and 10000 epochs.

between the generated image and the target domain image. The output image at the end of the proposed AtomGAN's training process is shown in Figure 6(j), and that for the CycleGAN model is shown in Figure 6(f). There are 7000 pixels at the peak in Figure 6(j); however, there are only 5000 pixels at the peak in Figure 6(f). Thus, the output image obtained by the proposed AtomGAN is closer to the gray distribution of the target segmentation result shown in Figure 6(b). In other words, the images generated by the proposed AtomGAN retain the basic atomic information of the input domain and better integrate the characteristics of the target domain. In addition, the proposed AtomGAN segments the defects effectively.

During the CycleGAN training process, we found that the loss of the discriminator was reduced too quickly, which resulted in the discriminator being accurate to distinguish the target domain from the generated pictures, and the loss of the generator did not converge. The generated picture is chaotic in the middle of the training (Figure 6(d)); thus, we stopped training the discriminator and trained only the generator. As a result, the generator's ability was improved. However, from 7000–10000 cycles, the generated image remained the same. Although a line defect was identified, the point defects were identified as line defects, and lines appeared in many places on normal atoms. According to the training output of the proposed AtomGAN, as the number of epochs increased, the style of the generated image gradually approached the target domain, and other features of the original image (e.g., the location information of atoms) were retained. During training, no additional artificial adjustments were needed. We found that the generator and discriminator were trained coherently using the total loss (Eq. (5)).

In the CycleGAN model, only the discriminator accepts the image of the target domain, which results in the training process becoming heavily dependent on the result of the discriminator. Once the discriminator becomes too powerful, the loss of the generator stops reducing; thus, the generator cannot identify the direction of generating the target domain image. In the proposed AtomGAN, we introduce the class encoder, and the class encoder branch extracts the target domain's features and integrates them into the coding part of the generator. Thus, the generator contacts the image of the target domain when generating the image, which reduces the over-dependence on the discriminator's result. This technique

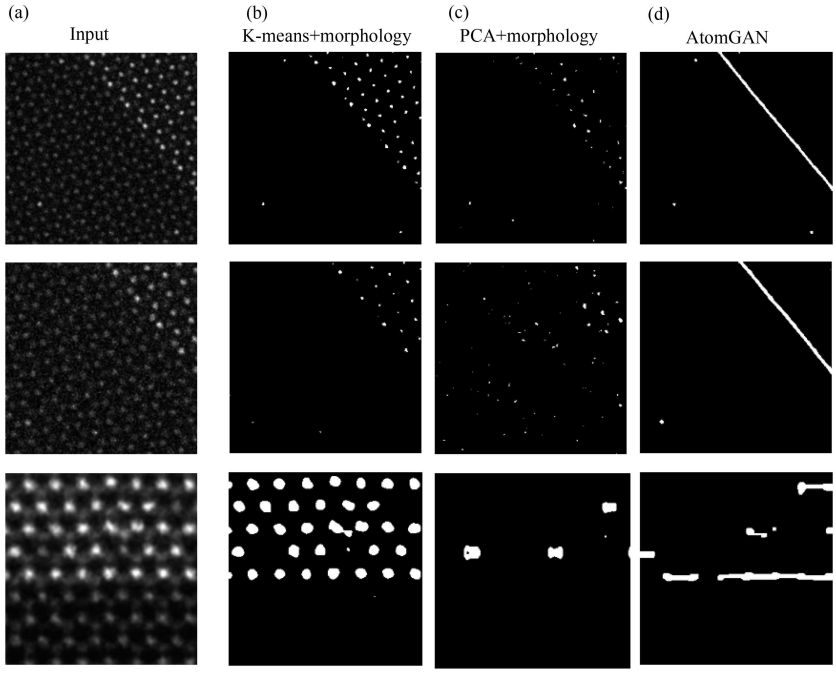


**Figure 7** Comparison with K-means and PCA algorithm. Comparison to other methods. (a) Inputs of algorthiums. The third row in (a) is from [29] Copyright 2014 Nature. (b)–(d) are results from K-means (with morphology), PCA (with morphology), and the AtomGAN (ours).

makes the generator training more directional and automatically balances the abilities of the generator and discriminator. Therefore, point and line defects can be identified successfully after training.

### 3.2 Segmentation results compared with other methods

The results of common segmentation algorithms were compared to those of the proposed model (Figure 7). It is difficult for these methods to perform well without human intervention, and they are extremely sensitive to noise. The first two images in Figure 7(a) are simulated STEM images at different scales, and the third image is a real STEM image of the in-plane interface between $WS_2$ and $MoS_2$ [29].

The K-means algorithm is a common cluster-based method [30]. The K-means method randomly selects K cluster centers and calculates the Euclidean distance (i.e., the difference of the pixel point value in the feature space) from the K centers to each pixel. Here, the values of pixels are divided into the K-th closest categories. Then, the center is reselected until the new center is equal to the original center. However, this method suffers from the hard-determined K-value and a random center point, which leads to different results. In addition, we found that the generated images (Figure 7(b)) were more focused on the background and foreground, ignoring the differences in the structure of the crystals because the algorithm only focuses on the pixel values of the image rather than the shapes of microstructures.

Rather than determining the class number artificially, principal component analysis (PCA) [31] decomposes the original pixel values into eigenvectors arranged according to the value of eigenvalues using nonlinear equations. The eigenvalue reflects the variance corresponding to the projection of pixels to the transformed axis. Here, a smaller eigenvalue indicates that the pixels are concentrated on this axis, and it is easier for defects and noise to have a larger offset on this axis. With PCA, appropriate eigenvector selection is a crucial but difficult step in defect detection because the perfect unit cell, defect, and noise overlap even after the transformation; thus, it is difficult to select an effective threshold to detect defects (Figure 7(c)).

It turns out that no other methods yield optimal results without tedious manual interventions because the shape routine must be readjusted when images have different atom scales. In addition, the results are sensitive to noise because the cutoff pixels' values are difficult to determine when the noise has a similar brightness as the atoms. Traditional image algorithms only extract simple features, e.g., texture and brightness features, and such algorithms also require expert prior knowledge. Thus, it is difficult for these methods to obtain all of the information of the image. The proposed AtomGAN benefits from

**Table 4** Confusion Matrix of predicted results

| | | Prediction | |
|---|---|---|---|
| | | Defect | Not defect |
| Ground truth | Defect | 3111 | 269 |
| | Not defect | 101 | 329659 |

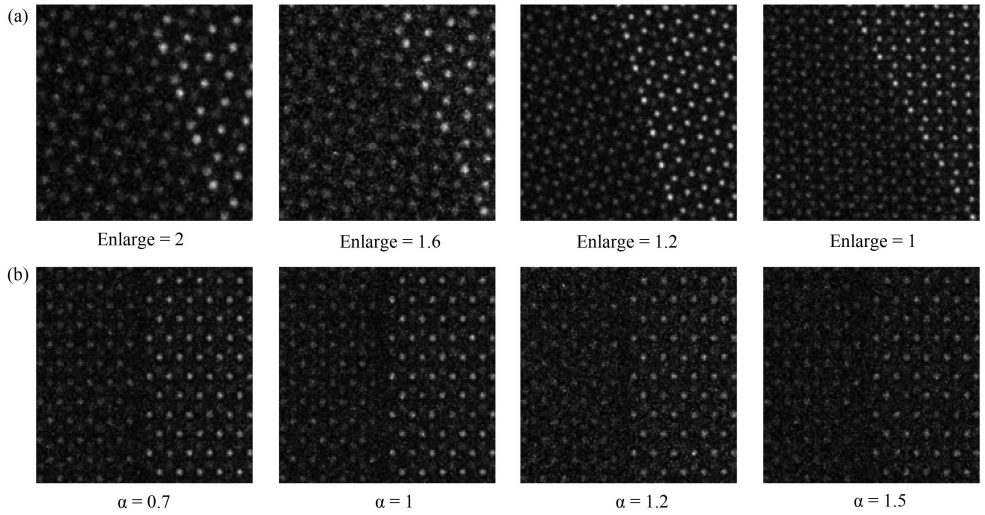


**Figure 8** Generated images with different scales and noise. (a) The generated images with different magnifications. Enlarge represents the magnification of the rightmost picture of (a). (b) The generated images with varying degrees of noise. $\alpha$ represents the degree of noise addition from (1).

multilayer CNNs that can automatically extract the high-level semantic features of images, e.g., contour, noise, and object features. Thus, AtomGAN can easily divide noises, defects, and target atoms and realize defect visualization and quantitative analysis, as shown in Figure 7(d). Note that the proposed AtomGAN performs well even with real electron microscopy imaging (third row in Figure 7), by using only using simulation datasets for training. This opens a novel opportunity for GANs to detect defects.

### 3.3 Evaluation of AtomGAN

The successful detection of defects is evaluated in terms of precision, recall, and F1 score, which are calculated according to the number of true positives, false positives, true negatives, and false negatives. The specific calculation method is as follows:

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}}, \\ \text{F1} &= 2\cdot\frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}}. \end{aligned} \tag{8}$$

To demonstrate the high performance of the proposed AtomGAN, we show the results of these metrics obtained on simulated datasets containing both point and line defects in Table 4. We found that the proposed model performed well even for atom images at different scales with different noise characteristics. The proposed model achieved a precision value of 96.9% and a recall value of 92.0% for defect detection. The F1 score obtained by the proposed model was 94.4%.

### 3.4 Results of generating defective samples

The cycled structure of the generator allows the images of both domains to be obtained (Figure 8). For the simulated STEM images, the segmented defect images are generated by the model, and the corresponding STEM images are generated by inputting the segmented defect images. The magnification of the generated electron micrograph can be determined by the input domain as shown in Figure 8(a), and

the degree of noise addition can be determined by the reference from the target domain as demonstrated in Figure 8(b). Thus, the type of electron microscope images generated can be manipulated. In addition, the images generated in this manner only require an average processing time of 0.07 s per image. Thus, the proposed model has the potential to expand the dataset used.

## 4 Conclusion

In this paper, an unsupervised learning model related to atom-scale defect detection was explored using the proposed AtomGAN model to segment both line and point defects simultaneously. Two types of unpaired simulation datasets were used to train the network, i.e., a simulated electron microscope photo dataset, and a defect segmentation image dataset. The proposed model implements a unique class encoder branch that avoids non-convergence loss during GAN training and makes the output convert to the target domain quickly. In addition, the network can generate both segmented images with accurate defect locations and the corresponding STEM images. The proposed AtomGAN model was evaluated experimentally, and the results demonstrated that the network performs well even with real electron microscope images. The AtomGAN provides a promising paradigm for the application of GANs in electron microscopy and can be applied in other fields that lack access to pairing datasets.

**Acknowledgements** This work was supported by National Key R&D Program of China (Grant No. 2022YFB2404303), National Natural Science Foundation of China (Grant Nos. 52107224, 52077096), State Grid Corporation of China (Grant No. 520626210064), and China Postdoctoral Science Foundation (Grant No. 2019M662612).

**References**

1 Han R, Feng S, Sun D M, et al. Properties and photodetector applications of two-dimensional black arsenic phosphorus and black phosphorus. Sci China Inf Sci, 2021, 64: 140402
2 Wu S, Wang X, Jiang W, et al. Interface engineering of ferroelectric-gated $MoS_2$ phototransistor. Sci China Inf Sci, 2021, 64: 140407
3 Yu X, Ng S, Putri L K, et al. Point-defect engineering: leveraging imperfections in graphitic carbon nitride (g-$C_3N_4$) photocatalysts toward artificial photosynthesis (small 48/2021). Small, 2021, 17: 2170252
4 Fu L, Li H, Wang L, et al. Defect passivation strategies in perovskites for an enhanced photovoltaic performance. Energy Environ Sci, 2020, 13: 4017–4056
5 Sangwan V K, Jariwala D, Kim I S, et al. Gate-tunable memristive phenomena mediated by grain boundaries in single-layer $MoS_2$. Nat Nanotech, 2015, 10: 403–406
6 Sangwan V K, Lee H S, Bergeron H, et al. Multi-terminal memtransistors from polycrystalline monolayer molybdenum disulfide. Nature, 2018, 554: 500–504
7 An J, Sun T, Wang B, et al. Efficient graphene in-plane homogeneous p-n-p junction based infrared photodetectors with low dark current. Sci China Inf Sci, 2021, 64: 140403
8 Wu R, Zhu R Z, Zhao S H, et al. Filling the gap: thermal properties and device applications of graphene. Sci China Inf Sci, 2021, 64: 140401
9 Yang Y, Zhou J, Zhu F, et al. Determining the three-dimensional atomic structure of an amorphous solid. Nature, 2021, 592: 60–64
10 Maksov A, Dyck O, Wang K, et al. Deep learning analysis of defect and phase evolution during electron beam-induced transformations in $WS_2$. npj Comput Mater, 2019, 5: 1
11 Ziatdinov M, Dyck O, Maksov A, et al. Deep learning of atomically resolved scanning transmission electron microscopy images: chemical identification and tracking local transformations. ACS Nano, 2017, 11: 12742–12752
12 Guo Y, Kalinin S V, Cai H, et al. Defect detection in atomic-resolution images via unsupervised learning with translational invariance. npj Comput Mater, 2021, 7: 180
13 Cho P, Wood A, Mahalingam K, et al. Defect detection in atomic resolution transmission electron microscopy images using machine learning. Mathematics, 2021, 9: 1209
14 Choi Y, Choi M, Kim M, et al. StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, 2018. 8789–8797
15 Jain S, Seth G, Paruthi A, et al. Synthetic data augmentation for surface defect detection and classification using deep learning. J Intell Manuf, 2020, 33: 1007–1020
16 Ren X, Lin W, Yang X, et al. Data augmentation in defect detection of sanitary ceramics in small and non-i.i.d datasets. IEEE Trans Neural Netw Learn Syst, 2022, : 1–10
17 Lian J, Jia W, Zareapoor M, et al. Deep-learning-based small surface defect detection via an exaggerated local variation-based generative adversarial network. IEEE Trans Ind Inf, 2020, 16: 1343–1351
18 Cannizzaro D, Varrella A G, Paradiso S, et al. In-situ defect detection of metal additive manufacturing: an integrated framework. IEEE Trans Emerg Top Comput, 2022, 10: 74–86
19 Tang T W, Kuo W H, Lan J H, et al. Anomaly detection neural network with dual auto-encoders GAN and its industrial inspection applications. Sensors, 2020, 20: 3336
20 Madsen J, Susi T. The abTEM code: transmission electron microscopy from first principles. Open Res Europe, 2021, 1: 24
21 Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV), Venice, 2017. 2242–2251
22 Zhu J Y, Zhang R, Pathak D, et al. Toward multimodal image-to-image translation. In: Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS), Long Beach, 2017

23 Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Proceedings of the 28th Conference on Neural Information Processing Systems (NIPS), Montreal, 2014. 2672–2680
24 Senel H G, Peters R A, Dawant B. Topological median filters. IEEE Trans Image Process, 2002, 11: 89–104
25 Huh W, Lee D, Lee C. Memristors based on 2D materials as an artificial synapse for neuromorphic electronics. Adv Mater, 2020, 32: 2002092
26 Lv Z, Xing X, Huang S, et al. Self-assembling crystalline peptide microrod for neuromorphic function implementation. Matter, 2021, 4: 1702–1719
27 He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks. In: Proceedings of European Conference on Computer Vision, 2016
28 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016. 770–778
29 Gong Y, Lin J, Wang X, et al. Vertical and in-plane heterostructures from $WS_2$/$MoS_2$ monolayers. Nat Mater, 2014, 13: 1135–1142
30 Bhola R, Krishna N H, Ramesh K N, et al. Detection of the power lines in UAV remote sensed images using spectral-spatial methods. J Environ Manage, 2018, 206: 1233–1242
31 Chaouch H, Charfeddine S, Aoun S B, et al. Multiscale monitoring using machine learning methods: new methodology and an industrial application to a photovoltaic system. Mathematics, 2022, 10: 890