Fully Automated Identification of Two-Dimensional Material Samples

Greplova, Eliska ; Gold, Carolin ; Kratochwil, Benedikt ; Davatz, Tim ; Pisoni, Riccardo ; Kurzmann, Annika ; Rickhaus, Peter ; Fischer, Mark H ; Ihn, Thomas ; Huber, Sebastian D

Abstract: Thin nanomaterials are key constituents of modern quantum technologies and materials research. The identification of specimens of these materials with the properties required for the development of state-of-the-art quantum devices is usually a complex and tedious human task. In this work, we provide a neural-network-driven solution that allows for accurate and efficient scanning, data processing, and sample identification of experimentally relevant two-dimensional materials. We show how to approach the classification of imperfect and imbalanced data sets using an iterative application of multiple noisy neural networks. We embed the trained classifier into a comprehensive solution for end-to-end automatized data processing and sample identification.

DOI: https://doi.org/10.1103/physrevapplied.13.064017

Fully Automated Identification of Two-Dimensional Material Samples

Eliska Greplova,[1,*] Carolin Gold,[2] Benedikt Kratochwil,[2] Tim Davatz,[2] Riccardo Pisoni,[2] Annika Kurzmann,[2] Peter Rickhaus,[2] Mark H. Fischer,[1,3] Thomas Ihn,[2] and Sebastian D. Huber[1]

[1] *Institute for Theoretical Physics, ETH Zurich, CH-8093 Zurich, Switzerland*
[2] *Solid State Physics Laboratory, ETH Zurich, CH-8093 Zurich, Switzerland*
[3] *Department of Physics, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland*

(Received 22 December 2019; revised manuscript received 8 April 2020; accepted 28 April 2020; published 8 June 2020)

Thin nanomaterials are key constituents of modern quantum technologies and materials research. The identification of specimens of these materials with the properties required for the development of state-of-the-art quantum devices is usually a complex and tedious human task. In this work, we provide a neural-network-driven solution that allows for accurate and efficient scanning, data processing, and sample identification of experimentally relevant two-dimensional materials. We show how to approach the classification of imperfect and imbalanced data sets using an iterative application of multiple noisy neural networks. We embed the trained classifier into a comprehensive solution for end-to-end automatized data processing and sample identification.

DOI: 10.1103/PhysRevApplied.13.064017

I. INTRODUCTION

Since the isolation of graphene [1], two-dimensional (2D) materials have constituted an active area of research, with numerous applications in optoelectronics [2–5] and as basic building blocks for a wide range of quantum devices [6]. This is due to the van der Waals stacking technique [7–9], which allows for drastic modifications of the band structure by stacking different materials, varying the number of layers, or introducing a twist between the layers [10–12].

High-quality van der Waals devices consist of flakes that are typically prepared by mechanical exfoliation [1]. Suitable flakes are identified via visual inspection in an optical microscope [13]. The shape, size, homogeneity, and thickness determine whether the flake is suitable for further processing. However, the observed difference in contrast and color of a flake with respect to the background not only depends on its thickness and material but also on the substrate that is used and on the settings of the microscope. This large parameter space makes the identification of usable flakes tedious and, while there exist proposed algorithmic solutions [14–21], a sufficiently general and fast algorithm is difficult to formulate. So far, many existing algorithmic approaches have concentrated on rule-based image processing [16] and a combination of the latter with machine learning [15,17]. While these methods are successful in the specific conditions of the respective study, it may be hard to generalize them to different experimental conditions, such as, e.g., various camera settings, illumination conditions, or substrates. Further approaches to the identification of suitable flakes comprise the automated classification and evaluation of previously collected flakes either by using optical-spectroscopy techniques [18,19] or via machine-learning methods [14,20–22].

Machine learning has proven to be a successful method for addressing the classification of large noisy data sets in many areas both in science and engineering [23–27]. In the present work, we address the issue of fast and reliable identification of 2D material samples using a supervised machine-learning algorithm. In this context, the problem of 2D sample identification is particularly daunting due to the lack of data (suitable flakes are, in general, rare and hard to find). Moreover, the challenge we are addressing here is not only to classify suitable flakes but also to formulate an end-to-end algorithm that enables scanning of the samples, preprocessing, fast classification, and identification of the suitable flakes in the frame of reference of the original sample. Once a suitable sample is identified, machine learning can also be used to further characterize the properties of the 2D flakes in real time, as recently demonstrated in Ref. [22].

Dealing with realistic data sets can overwhelm even established machine-learning methods, as the collected data can be insufficient, unbalanced, mislabeled, and can have a high amount of noise. Here, we first discuss the efficient collection of suitable hexagonal boron-nitride (hBN) flakes, before we extend our approach to both graphite and graphene flakes. The search for 2D material flakes is an example of the broader class of noisy classification

*geliska@phys.ethz.ch

problems: high-quality 2D materials are crucial elements of many quantum device experiments but their collection is difficult, as appropriate flakes are exceedingly rare and qualitatively very different from each other. Finding 2D material flakes typically requires many hours of expert human labor or advanced image-processing software. The ideal flake is uniform, isolated from other flakes, and has a certain thickness that is suited for a specific experiment. For example, hBN is typically used as either a flat substrate (in this case, the thickness is not relevant) or as a gate dielectric (with a thickness between 10 and 90 nm). In addition, different users may prefer slightly different flakes. These challenges, however, make machine learning, and particularly neural networks, highly apt for this task: the goal of machine learning is to extract general features from a limited set of data. An additional advantage of machine-learning methods is that, once trained, the model can be applied to new data in a matter of seconds, as opposed to the repeated run of an computationally heavy algorithm.

In this paper, we present a fully automated solution for identifying suitable 2D material flakes on a wafer. We automatize a setup consisting of a microscope, a camera, and a sample stage in a glove box (for further details, see Appendix B) to scan wafers carrying manually exfoliated flakes. In a preprocessing step, flakes on the wafer are detected and uniformly formatted. A set of multilayer convolutional neural networks is employed to identify promising flakes. In a last step, the algorithm captures pictures of the promising flakes in a higher resolution. We report a success rate of our algorithm that is comparable to that of a human operator, while being significantly faster.

II. ALGORITHM

A human operator bases their classification of 2D material flakes into "good" or "bad" flakes using various factors. The size, shape, homogeneity, the amount of bubbles, and the color (which relates to the thickness) are considered. For example, good flakes of hBN are blue to yellow on our substrate, homogeneous, and large. These criteria are not strictly objective and there is a significant user-to-user variation in the classification.

Deep neural networks are the state-of-the-art candidates for solving diverse classification problems [28–34]. Our algorithm utilizes a neural network to classify 2D flakes on the fly. While the microscope automatically scans the wafer, the algorithm processes the pictures and returns the coordinates of the experimentally relevant flakes on the sample. In the end, the human operator obtains a small selection of flakes that are labeled as "good" by the algorithm.

Our approach to collecting and classifying data consists of the following steps: (a) automated data collection ("scanning"); (b) manual labeling of the obtained data; (c) preprocessing of the data; (d) training of the machine-learning model; (e) applying the trained neural network to new data sets; and (f) identifying the coordinates of the flakes of interest on the sample. Steps (b) and (d) are only performed once and are skipped when the trained model is applied to new raw data sets.

Scanning. Step (a) is accomplished by an automated setup consisting of an off-the-shelf microscope, a camera, and a sample stage in a glove box. This setup scans wafers with manually exfoliated 2D flakes on their surface and captures an image at fixed positions. Prior to the scan, the scanning area and focal plane of the sample are defined through user input. This step takes minutes and, crucially, is independent of the wafer size. At each position, the camera automatically adjusts the focus according to the user input and captures an image. The machine-learning algorithm explained below processes these images in parallel to the scanning, thus avoiding any downtime of the setup. At a microscope magnification of $10\times$, scanning a 1 cm^2 of flake material takes 190 s. This is significantly faster than for the average human operator. An example of a typical image of hBN obtained by sample scanning is shown in Fig. 1(a).

Labeling. In step (b), the images obtained in step (a) are labeled by human operators who identify suitable flakes. The labeling is done by different human operators and each flake is only looked at once by a single operator. This labeled data is used for training the machine-learning algorithm. The labeling and training step is only performed once. To facilitate the labeling process, we develop a graphical user interface, where the user simply clicks on the samples they considers suitable for further processing. The coordinates of the flakes within the given picture, as well as within the whole wafer, are saved.

Preprocessing. Having collected (and, in the case of the training phase of the algorithm, labeled) the data, the preprocessing step (c) is applied. The minimal preprocessing algorithm identifies potential candidates for usable flakes and formats them uniformly for the classification step. We use the PYTHON Image Library (PIL) to process the data and check it for large standard deviation. Whenever the standard deviation exceeds a chosen threshold, the algorithm cuts out a fixed-size square around the given point. The square images (80×80 pixels) then become elements of the training or evaluation set for the algorithm. At this stage, every flake is uniquely identified by its coordinates with respect to the coordinate system spanned by three corners for the wafer. This identifier is kept throughout all further steps. Examples of suitable hBN flakes are shown in the right-hand column of Fig. 1(b), while examples of unsuitable flakes are depicted in the left-hand column.

We use the preprocessing part of the algorithm to assign two labels to the training images. When the square image created around an interesting point in the data contains a

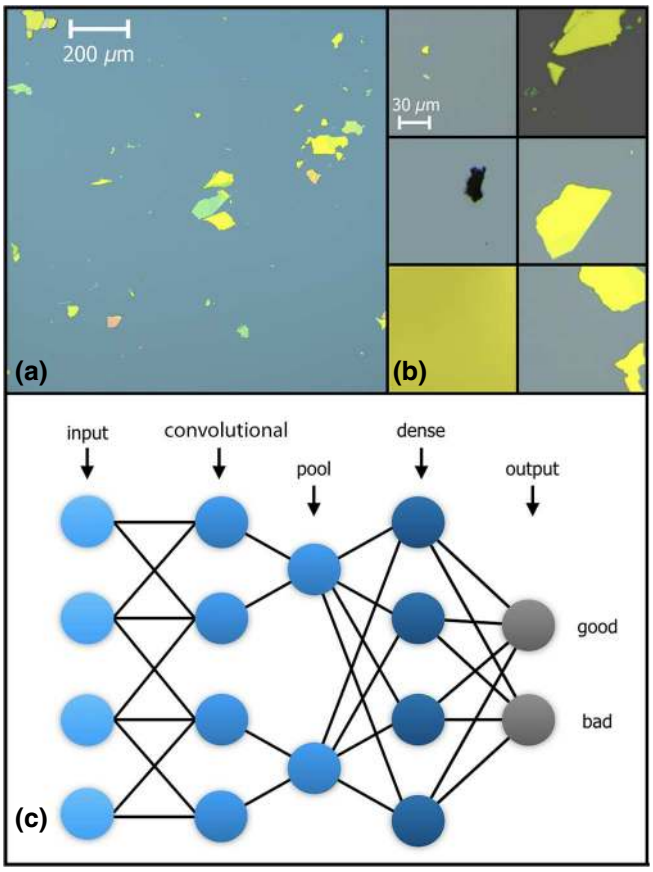

FIG. 1. The collection and processing of data. (a) A typical microscope image of hBN. (b) Examples of the preprocessed images: the first column depicts examples picked up by our preprocessing algorithm that are not hBN flakes, whereas the second column shows hBN flakes labeled as "good". (c) A schematic illustration of the architecture of the neural network that is used: the convolutional layers are followed by pooling and are connected to the dense layer—the two-neuron output layer outputs probabilities for a given flake being suitable ("good") or unsuitable ("bad") for further processing.

suitable flake, as identified in step (b), the image is labeled as "good"; otherwise, it is labeled as "bad." The resulting data set contains a large number of unsuitable ("bad") flakes identified by the standard-deviation filter, since the threshold criterion has to be sufficiently loose such that no good flake will be missed, independent of its size, thickness, or the color of the wafer.

We artificially increase the number of "good" flakes in the training set by using image rotations and mirror reflections, such that each "good" flake enters the training set in six different variations. This choice significantly shortens the preparation of the training set, i.e., the training set will, with less effort, contain a sufficient amount of "good" flakes for successful training of our model.

As seen in Fig. 1(b), the examples of both "good" and "bad" flakes are very diverse. On the one hand, the flakes of various thicknesses and shapes have to be recognized as good even against the background, the color of which can vary depending on the wafer. On the other, flakes that are too small, broken, or simply just dust or pictures taken outside of the wafer all have to be identified as "bad." This diversity of the different flakes has immediate consequences for the construction of the classification mechanism.

Training. In step (d), a reliable binary classification model is trained using the training set from step (c). This classifier is trained to distinguish two classes of flakes: "good" and "bad."

The architecture of the network in use is shown in Fig. 1(c). We employ a deep network consisting of four convolutional layers and one dense layer. The convolutional layers have 64, 64, 128, and 256 filters, respectively. The dense layer has 256 neurons (for further details, see Appendix A). The output of the network is the probability distribution between the two classes "good" and "bad" in the output layer. The weights for the neurons in the network are trained using standard back-propagation methods based on the training set [35].

The data set from step (c) is highly imbalanced, meaning that the number of flakes with the label "good" is very low compared to the number of flakes with the label "bad." In general, such an imbalanced data set for training will result in a network that labels all the flakes with the label "bad" and not learning any feature of the flakes with the label "good." On the other hand, down-sampling the number of "bad" flakes will lead to a loss of the variability of the data set. We overcome these issues by forming small balanced sets, so-called batches, out of the labeled data and by creating an iterative protocol consisting of multiple neural networks applied consecutively. In particular, in each training step, the network is fed a batch that consists of randomly selected 50% "good" and 50% "bad" flakes. To avoid overfitting the networks, the training is interrupted while the network is still relatively noisy and has an accuracy of around 90%. At this point, we train further networks separately on the same training data. In our case, we find that applying three separate models is optimal. If a given flake consecutively passes the three separately trained noisy classifiers, there is an increased probability that it is worth being inspected by a human expert. The scheme of the iterative model is illustrated in Fig. 2. When generalizing to other materials, the number of networks needed may differ depending on the particular distribution of good flakes in the training set.

Application of the model. Once the model is trained, we can, in step (e), apply it to any preprocessed data set that was not part of the original training data. The unique coordinates of each flake identified as "good" by the algorithm allow us to easily navigate to any flake under the microscope and to use it for further processing or device fabrication.

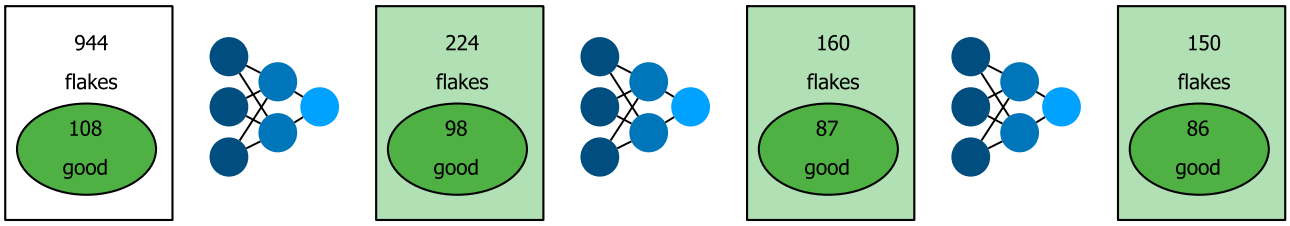

FIG. 2. The application of the multiple networks. The iterative procedure filters out bad flakes on the concrete example of a small set of 944 flakes. The rectangle on the left of each network is its input and the rectangle on the right of the network is its number of flakes classified by the network to be good. The dark green ovals denote the number of good flakes present in each step according to the human operator's labeling. The input of the algorithm is 944 flakes, out of which 108 are labeled as good by a human operator. The first iteration of the networks classifies 224 flakes as suitable for further processing. Out of these, 98 are actually labeled as good by a human expert. Therefore, the network misses ten good flakes in this first step. In the second step of the iterative algorithm, we input the 224 flakes selected by our first model into the second neural network. The second network classifies 160 flakes as suitable, out of which 87 are labeled as good by a human operator. In the second step, the model misses an additional 11 good flakes. Finally, we run the last neural network on the output of the second neural network. The model now selects 150 flakes as suitable, out of which 86 are labeled as good. In this last step, the algorithm misses one good flake from the set of good flakes identified in the previous step.

III. RESULTS

We first discuss the results for hBN on a Si substrate. For this material, each of the three networks is trained on approximately 10^5 flakes. Of these, about 10^4 are labeled as good. For training, we use batches of approximately 200 flakes each and perform 1.2×10^5 training steps. We evaluate the performance of the model every 1000 training steps on a fixed evaluation batch consisting of 10% of the total number of "good" flakes and an equal number of "bad" flakes.

The evaluation loss and accuracy of all models are shown in Figs. 3(a)–3(c). The evaluation loss is the cross-entropy between the label ("good" or "bad") predicted by the network and the label determined by an expert. The accuracy is the percentage of correctly identified flakes within the chosen evaluation batch (the evaluation batch is different for the different neural networks). The accuracy reached at the end of the training is between 87% and 88%, as shown in Fig. 3. While the loss for the trained models keeps decreasing toward zero, the training is intentionally finished at its nonzero finite value, as explained in Sec. II (see also Appendix A).

The performance of the protocol for hBN is shown in Fig. 2. The first network is applied on a preprocessed data set of 944 flakes, out of which 108 are "good" (human operator). These data points are not seen by the neural network, i.e., are not part of the training set. Thus, the network does not have access to the labels. After the first application of the network, we obtain 224 flakes marked as "good" by the network. Out of these 224 flakes, 98 are "good" as determined by a human expert (this also implies that the network missed (assigned a false negative to) ten out of all 108 good flakes). We take the flakes marked by the network and use them as input for the second network (see Fig. 2) and so on. After the application of the third network, we obtain 150 flakes, out of which 86 are labeled as "good" by an expert. The application of the networks takes a few seconds and it puts the human operator into a situation in which approximately 55% of the flakes are suitable for further experimental use. Therefore, around 80% of the flakes marked as good by a human operator are reliably found by the algorithm, whereas the remaining 20% of the "good" flakes are missed.

To evaluate the success of the protocol on the new unlabeled data, we ask three human operators working on different 2D-material-based quantum devices to independently evaluate a previously unlabeled set of 100 hBN flakes that the network has selected as "good." The human operators denote 73, 51, and 45 flakes as "good." As the exact flake size, shape, and the thickness desired for the fabrication of quantum devices is strongly dependent on the exact application, we observe a large user-to-user variation. Within this context, it is more descriptive to look at how many flakes picked as "good" by the algorithm are denoted as "bad" by all human operators. In our case, only seven of the 100 test flakes are denoted as "bad" by all three experts. This is consistent with the benchmarks that we obtain from the validation procedure of our protocol, which leads us to expect that about 55% of the flakes picked by the algorithm are of further use. The human validation is illustrated in Fig. 4. One can observe a considerable variability in the human assessment of the flakes and very small amount of images that are considered to be mislabeled by all three operators. This shows that our way of labeling the flakes introduces sufficient diversity of criteria into the training set. Due to this diversity, the selected flakes are useful for multiple users with a range of different requirements.

To test the robustness of the algorithm and to ensure that it is not constrained to our specific setup, we test the algorithm in two separate glove-box systems. The setups have different cameras and light conditions and only the Si substrate used is kept the same. The automated

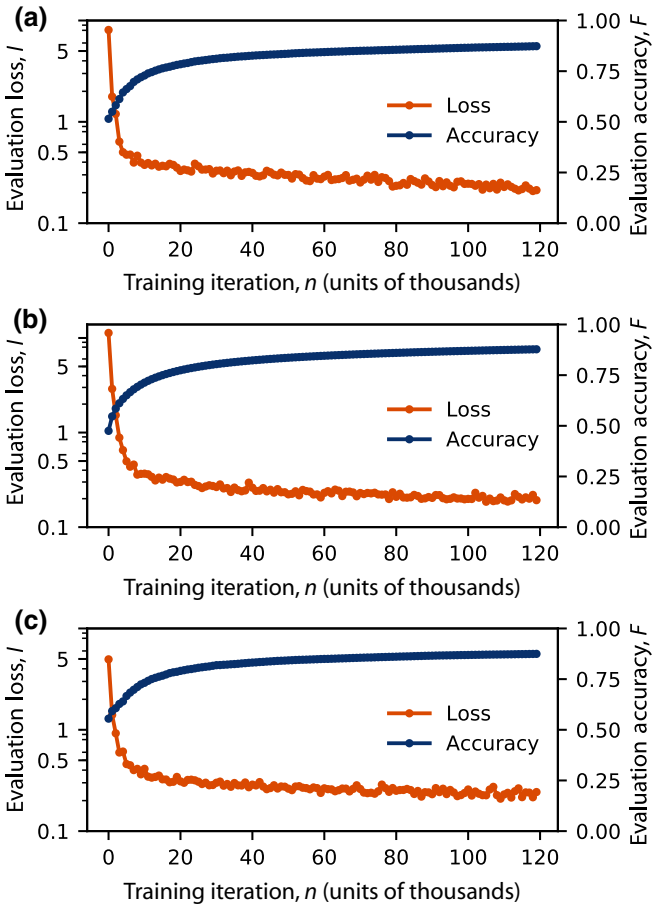

FIG. 3. The loss and accuracy of the three models used in the iterative algorithm. The networks are trained using a training set of 10^5 labeled flakes. Panels (a), (b), and (c) show the evaluation loss in orange and the corresponding evaluation accuracy in blue for the three models used in the algorithm as shown in Fig. 2.

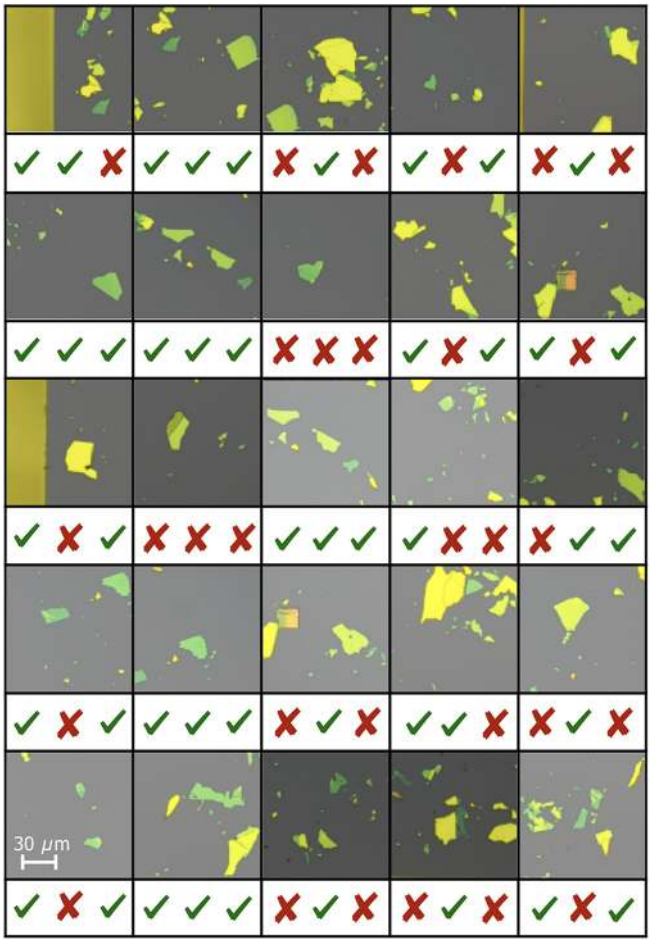

FIG. 4. Validation by a human operator. The first 25 flakes of the set for human validation are shown with their evaluation by three different operators: 23 flakes are denoted to be suitable by at least one operator, while two flakes are dismissed by all three operators. While the user-to-user variation is significant, the algorithm provides flakes that are mostly considered suitable by at least one operator.

identification of suitable hBN flakes in this new setup leads to results that are very similar to those described above, i.e., roughly every second flake is useful to an experimentalist. This gives us great confidence that our system is readily transferable to other laboratories featuring different instruments.

To test the flexibility of our algorithm with respect to different materials, we furthermore apply the algorithm to graphite flakes as well as graphene flakes. Graphite plays a similarly important role for quantum materials experiments as hBN. In Fig. 5, we show an example of a typical microscope picture of a graphite sample [panel (a)] and examples of "bad" and "good" flakes [panel (b)]. For graphite, flakes of the various shades of blue (thickness around 10–20 nm) are sought after, while the large yellow flakes (thickness 50 nm or more) are not desirable. To adjust for this new thickness (and therefore color preference), we retrain the models on approximately 200 manually labeled graphite flakes. Because the models are already trained for hBN and thus identify most features of useful flakes, we only need a small training set for the retraining process. We test the retrained network on exfoliated graphite flakes [see Fig. 5(c)] previously unseen by the algorithm. Our preprocessing algorithm identifies 1200 flakes, out of which 132 are "good" graphite flakes. The retrained model, however, determines 185 flakes to be good. Out of these, 109 flakes have been labeled as "good" by a human operator. In other words, we obtain a success rate of about 60%, which is comparable to the case in which the algorithm is applied to hBN flakes.

As a final challenge, we test our model on bilayer graphene, see Fig. 6. Bilayer-graphene flakes are one of the most challenging to identify because of their thinness and small size. We collect 48 bilayer-graphene flakes and use them to retrain our model. We then test the trained model

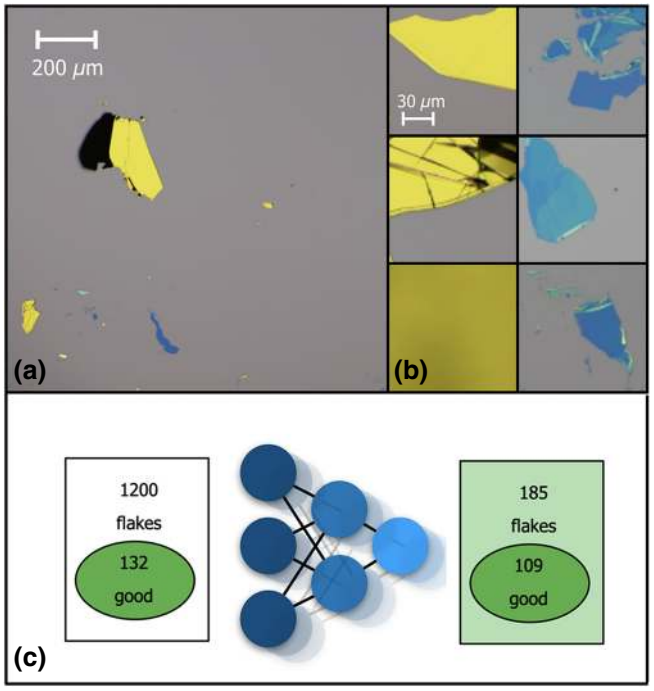

FIG. 5. Testing the performance of the network on different materials: (a) an example of a graphite wafer; (b) examples of bad (left) and good (right) graphite flakes; (c) the results of the application of our protocol to 1200 flakes obtained by scanning a single wafer. The test set of 1200 flakes contains 132 flakes labeled as "good" by a human operator. The three iterations of the protocol propose that 185 flakes are suitable, out of which 109 are actually good (as labeled by a human operator). This result shows that approximately 60% of the flakes suggested by the model are indeed suitable. In total, the network identifies 83% of the "good" flakes in the set while missing 17%.

on a separate test set of approximately 3800 flakes that contains 36 flakes labeled as "good" by human operators, i.e., around 1% of the flakes. Our model identifies 262 "good" flakes, out of which 26, i.e., 10%, are actually considered good by human experts. The algorithm therefore significantly increases the density of good flakes in the sample that a human expert has to go through, while at the same time only losing a small percentage of good flakes that are not picked up by the algorithm. While we do not achieve the same performance as that of the model on hBN and graphite, we succeed in increasing the occurrence of useful bilayer-graphene flakes by one order of magnitude, from around 1% to 10%. This achievement serves as a first step toward accumulating a large database of bilayer graphene, which will be a stepping stone toward new methods of automated graphene collection.

IV. DISCUSSION AND CONCLUSIONS

We present a fully automated method to evaluate the quality and suitability of 2D-material specimens for

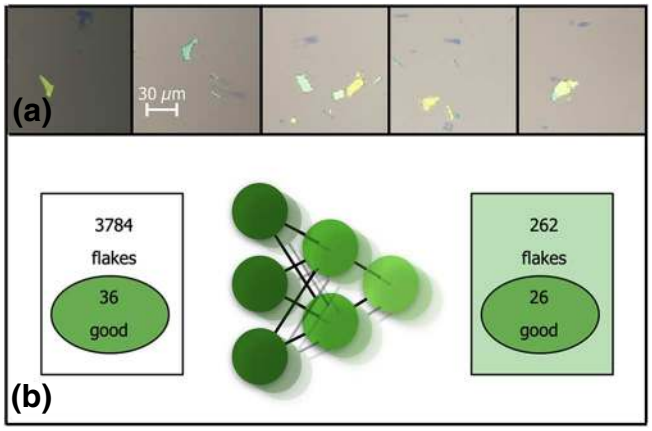

FIG. 6. Testing the performance of the network on bilayer graphene: (a) examples of preprocessed data containing bilayer-graphene flakes; (b) the results of the application of our protocol on 3800 flakes obtained by scanning a single wafer.

sample fabrication. With the inclusion of every step from data collection and preprocessing to the evaluation of suitable flakes, the algorithm removes 90% of the redundant data without human assistance. From the remaining 10% of the flakes labeled as "good" by the algorithm, more than half are actually suitable for further experimental use for the case of hBN and graphite. The exact percentage of useful flakes can vary depending on the needs of the respective application. For example, hBN flakes can be used for experiments with twisted bilayer graphene [36] and these experiments typically require thick (90-nm) and huge flakes (100 μm). On the other hand, samples used for nanodevices [37–39] require thin (about 20-nm) and small (20-μm) hBN flakes. Hence, almost all of the hBN flakes identified by the network are usable for one of our experiments, as seen in the human validation shown in Fig. 4.

We train and successfully test our algorithm on the detection of hBN, graphite, and bilayer-graphene flakes. In the case of hBN, we create a very robust training set containing thousands of good hBN flakes. The resulting model is reliable and can be easily translated for the identification of graphite flakes due to the similar quality characteristics of good hBN and graphite flakes. Transferring the model to the identification of bilayer graphene, however, is not so easy, as the differences in the quality features of good hBN or graphite flakes and bilayer-graphene flakes are larger. The retraining of our model to identify good bilayer-graphene flakes is furthermore hampered by the lack of a vast data set of bilayer-graphene flakes labeled as 'good'. Therefore, the accuracy for bilayer graphene is significantly lower than for hBN and graphite. Nonetheless, the algorithm is still successful in eliminating a very large percentage of unsuitable flakes and leaves a human operator with in the order of 100 flake candidates, out of

which around 10% are suitable for further processing. This procedure allows us to create a sufficiently large database of graphene flakes and train further neural-network models with higher success rates.

The extension of the algorithmic strategies presented here to other 2D materials and platforms promises to facilitate the fabrication of a large variety of different 2D-material-based devices [40–43]. The ready-to-use algorithm is available online [44]. We provide both API for the automation of the data collection as well as pretrained models that can be applied on newly collected raw data.

ACKNOWLEDGMENTS

We gratefully acknowledge financial support from the Swiss National Science Foundation, the National Centre of Competence in Research (NCCR)−Quantum Science and Technology (QSIT). This work has received funding from the European Research Council under Grant Agreement No. 771503. We acknowledge helpful discussions with Klaus Ensslin and Yuya Shimazaki. We also acknowledge Rebekka Garreis, Chuyao Tong, and Yongjin Lee for contributing to the labeled database. Finally, we thank Peter Blake for his help with the data acquisition from the microscope.

C. G. and B. K. contributed equally to this work.

Note added.—Recently, we became aware of a similar independent work by Masubuchi *et al.* [45].

APPENDIX A: NEURAL-NET ARCHITECTURE

Neural networks are nonlinear functions of many parameters (the weights and biases of each neuron) mapping input to output. The networks are trained such that a certain cost function between input and output is minimized. This minimization is achieved by optimizing the weights and biases of the neuron using the back-propagation method [35]. In this work, we use three neural networks separately trained on the same highly imbalanced training set consisting of approximately 10^5 samples. Roughly 10% of these samples correspond to flakes of 2D material that are labeled as "good," i.e., they are of further experimental use. Not all these approximately 10^4 flakes are original, most of them having been created by mirroring and rotations of the rare experimentally collected good flakes. We train on batches of 200 flakes (100 of them "good," 100 of them "bad"). As discussed in the main text, this balance is crucial for optimizing the network variables correctly. We train the network using 1.2×10^5 training steps. We evaluate the model every 1000 training steps on a fixed evaluation batch consisting of 10% of the total number of "good" flakes and an equal number of "bad" flakes. The accuracy at the end of the training is between 87% and 88%, as shown in Figs. 3(a)–3(c). The cost function that we aim to minimize by training is the cross-entropy between the label ("good" or "bad") determined by the network (y^{output}) and the true label assigned by an experimentalist (y^{target}), which can be expressed as

$$H(y^{\text{target}}, y^{\text{output}}) = -\sum_j y^{\text{target}} \ln y^{\text{output}}, \tag{A1}$$

where $y^{\text{target}}, y^{\text{output}} \in \{\text{good}, \text{bad}\}$. By trying to minimize Eq. (A1), we penalize the misclassification of both "good" and "bad" flakes and force the network to find the weights that lead to recovery of the correct classification label.

TABLE I. A summary of the hyperparameters of the neural net.

Type of layer	Filters	Kernel size	Strides	Neurons
Convolutional	64	5	2	NA
Convolutional	64	3	1	NA
Convolutional	128	3	1	NA
Convolutional	256	3	1	NA
Dense	NA	NA	NA	256

Let us describe the structure of our models in more detail. As mentioned above, each of the networks that we use consists of four convolutional layers and one dense layer. The hyperparameters for the layers that we use are summarized in Table I. The first convolutional layer is defined with a larger kernel and stride and a low amount of filters. The deeper layers contain a larger number of filters and smaller kernels. The dense layer contains 256 neurons. These neurons are then connected to two output neurons that contain a probability that the given flake is "good" or "bad," respectively. The activation function for all the layers except the last is a rectified linear unit (ReLU), while the last layer contains a softmax activation function that transforms the values of the neurons into the probability distribution. For a detailed overview of both the theoretical aspects and the practical implementation of neural networks, see Refs. [27,46].

We are using iterative applications of three independently trained models. The number of models to be applied is chosen by numerical optimization. Our main objective is to minimize the number of false positives ("bad" flakes labeled as "good") and at the same time the number of false negatives (the "good" flakes labeled as "bad"). These two conditions translate into the following requirements: the human operator should have minimal overhead when selecting the best flake for they purposes from the suggested set. But, at the same time, this overhead minimization should not lead to overwhelming loss of suitable flakes. The balance that we find leads to the network suggesting sets that contain about 60% of suitable flakes, while at most 20% of the good flakes are lost. Given the processing speed and the amount of flakes analyzed, this loss is not significant.

TABLE II. The confusion matrices for the evaluation of our model on the new data for hBN (left), graphite (center), and graphene (right). The rows represent the correct labels and the columns the labels assigned by the model.

	Good	Bad		Good	Bad		Good	Bad
Good	86	22	Good	109	23	Good	26	10
Bad	64	794	Bad	76	1015	Bad	236	3522

In Table II, we summarize the performance of the models for hBN, graphite, and graphene via the confusion matrices of their evaluation on the test sets shown in Figs. 2, 5, and 6.

APPENDIX B: EXPERIMENTAL SETUP

We test and run our algorithm in two separate glove-box setups. Below, we list the equipment used for the automated scanning for both setups.

Glove box 1:

(a) LV Econ Microscope with motorized z axis and motorized revolver
(b) Märzhauser Stage with Tango Desktop controller
(c) DS RI 2 Camera

Glove box 2:

(a) LV Econ Microscope with motorized z axis and motorized revolver
(b) Märzhauser Stage with Tango Desktop controller
(c) DS FI 3 Camera

[1] K. S. Novoselov, A Mishchenko, A Carvalho, and A. H. Castro Neto, 2D materials and van der Waals heterostructures, Science **353**, aac9439 (2016).

[2] Fengnian Xia, Han Wang, Di Xiao, Madan Dubey, and Ashwin Ramasubramaniam, Two-dimensional material nanophotonics, Nat. Photonics **8**, 899 (2014).

[3] F. Bonaccorso, Z. Sun, T. Hasan, and A. C. Ferrari, Graphene photonics and optoelectronics, Nat. Photonics **4**, 611 (2010).

[4] F. H. L. Koppens, T. Mueller, Ph. Avouris, A. C. Ferrari, M. S. Vitiello, and M. Polini, Photodetectors based on graphene, other two-dimensional materials and hybrid systems, Nat. Nanotechnol. **9**, 780 (2014).

[5] Qing Hua Wang, Kourosh Kalantar-Zadeh, Andras Kis, Jonathan N. Coleman, and Michael S. Strano, Electronics and optoelectronics of two-dimensional transition metal dichalcogenides, Nat. Nanotechnol. **7**, 699 (2012).

[6] K. S. Novoselov, A. Mishchenko, A. Carvalho, A. H. Castro Neto, and Oxford Road, 2D materials and van der Waals heterostructures, Science (80-.). **353**, aac9439 (2016).

[7] C. R. Dean, A. F. Young, I. Meric, C. Lee, L. Wang, S. Sorgenfrei, K. Watanabe, T. Taniguchi, P. Kim, K. L. Shepard, and J. Hone, Boron nitride substrates for high-quality graphene electronics, Nat. Nanotechnol. **5**, 722 (2010).

[8] A. K. Geim and I. V. Grigorieva, Van der Waals heterostructures, Nature **499**, 419 (2013).

[9] Riccardo Frisenda, Efrén Navarro-Moratalla, Patricia Gant, David Pérez De Lara, Pablo Jarillo-Herrero, Roman V. Gorbachev, and Andres Castellanos-Gomez, Recent progress in the assembly of nanodevices and van der Waals heterostructures by deterministic placement of 2D materials, Chem. Soc. Rev. **47**, 53 (2018).

[10] Yuan Cao, Valla Fatemi, Shiang Fang, Kenji Watanabe, Takashi Taniguchi, Efthimios Kaxiras, and Pablo Jarillo-Herrero, Unconventional superconductivity in magic-angle graphene superlattices, Nature **556**, 43 (2018).

[11] Matthew Yankowitz, Shaowen Chen, Hryhoriy Polshyn, Yuxuan Zhang, K. Watanabe, T. Taniguchi, David Graf, Andrea F. Young, and Cory R. Dean, Tuning superconductivity in twisted bilayer graphene, Science **363**, 1059 (2019).

[12] Hiram J. Conley, Bin Wang, Jed I. Ziegler, Richard F. Haglund, Jr., Sokrates T. Pantelides, and Kirill I. Bolotin, Bandgap engineering of strained monolayer and bilayer MoS_2, Nano Lett. **13**, 3626 (2013).

[13] P. Blake, E. W. Hill, A. H. Castro Neto, K. S. Novoselov, D. Jiang, R. Yang, T. J. Booth, and A. K. Geim, Making graphene visible, Appl. Phys. Lett. **91**, 063124 (2007).

[14] Yuhao Li, Yangyang Kong, Jinlin Peng, Chuanbin Yu, Zhi Li, Penghui Li, Yunya Liu, Cun-Fa Gao, and Rong Wu, Rapid identification of two-dimensional materials via machine learning assisted optic microscopy, J. Materiomics **5**, 413 (2019).

[15] Satoru Masubuchi and Tomoki Machida, Classifying optical microscope images of exfoliated graphene flakes by data-driven machine learning, npj 2D Mater. Appl. **3**, 4 (2019).

[16] Satoru Masubuchi, Masataka Morimoto, Sei Morikawa, Momoko Onodera, Yuta Asakawa, Kenji Watanabe, Takashi Taniguchi, and Tomoki Machida, Autonomous robotic searching and assembly of two-dimensional crystals to build van der Waals superlattices, Nat. Commun. **9**, 1413 (2018).

[17] S. Funke, U. Wurstbauer, B. Miller, A. Matković, A. Green, A Diebold, C. Röling, and P. H. Thiesen, Spectroscopic imaging ellipsometry for automated search of flakes of mono- and n-layers of 2D-materials, Appl. Surf. Sci. **421**, 435 (2017).

[18] Roman V. Gorbachev, Ibtsam Riaz, Rahul R. Nair, Rashid Jalil, Liam Britnell, Branson D. Belle, Ernie W. Hill, Kostya S. Novoselov, Kenji Watanabe, Takashi Taniguchi, Andre K. Geim, and Peter Blake, Hunting for monolayer boron nitride: Optical and Raman signatures, Small **7**, 465 (2011).

[19] Hai Li, Jumiati Wu, Xiao Huang, Gang Lu, Jian Yang, Xin Lu, Qihua Xiong, and Hua Zhang, Rapid and reliable thickness identification of two-dimensional nanosheets using optical microscopy, ACS Nano **7**, 10344 (2013).

[20] Xiaoyang Lin, Zhizhong Si, Wenzhi Fu, Jianlei Yang, Side Guo, Yuan Cao, Jin Zhang, Xinhe Wang, Peng Liu, Kaili Jiang, and Weisheng Zhao, Intelligent identification of two-dimensional nanostructures by machine-learning optical microscopy, Nano Res. **11**, 6316 (2018).

[21] Yu Saito, Kento Shin, Kei Terayama, Shaan Desai, Masaru Onga, Yuji Nakagawa, Yuki M. Itahashi, Yoshihiro Iwasa, Makoto Yamada, and Koji Tsuda, Deep learning-based quality filtering of mechanically exfoliated 2D crystals, arXiv:1907.03239, 2019.

[22] Bingnan Han, Yuxuan Lin, Yafang Yang, Nannan Mao, Wenyue Li, Haozhe Wang, Valla Fatemi, Lin Zhou, Joel I-Jan Wang, Qiong Maa, *et al.*, Deep learning enabled fast optical characterization of two-dimensional materials, arXiv:1906.11220 , 2019.

[23] Sotiris B. Kotsiantis, I. Zaharakis, and P. Pintelas, Supervised machine learning: A review of classification techniques, Emerg. Artif. Intell. Appl. Comput. Eng. **160**, 3 (2007).

[24] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G. R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, Phys. Rep. **810**, 1 (2019).

[25] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, New Jersey, US, 2018), p. 8697.

[26] Jie Hu, Li Shen, and Gang Sun, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, New Jersey, US, 2018), p. 7132.

[27] Michael A. Nielsen, *Neural Networks and Deep Learning* (Determination Press, San Francisco, CA, 2015), Vol. 25.

[28] Stephan Dreiseitl and Lucila Ohno-Machado, Logistic regression and artificial neural network classification models: A methodology review, J. Biomed. Inform. **35**, 352 (2002).

[29] Eric A. Wan, Neural network classification: A Bayesian interpretation, IEEE Trans. Neural Networks **1**, 303 (1990).

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, in *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, Massachusetts, US, 2012), p. 1097.

[31] Dan Cireşan, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber, Multi-column deep neural network for traffic sign classification, Neural Netw. **32**, 333 (2012).

[32] Juan Carrasquilla and Roger G. Melko, Machine learning phases of matter, Nat. Phys. **13**, 431 (2017).

[33] Eliska Greplova, Christian Kraglund Andersen, and Klaus Mølmer, Quantum parameter estimation with a neural network, arXiv:1711.05238 , 2017.

[34] R. Durrer, B. Kratochwil, J.V. Koski, A.J. Landig, C. Reichl, W. Wegscheider, T. Ihn, and E. Greplova, Automated Tuning of Double Quantum Dots into Specific Charge States Using Neural Networks, Phys. Rev. Applied **13**, 054019 (2020).

[35] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, Learning representations by back-propagating errors, Cognit. Modell. **5**, 1 (1988).

[36] Peter Rickhaus, John Wallbank, Sergey Slizovskiy, Riccardo Pisoni, Hiske Overweg, Yongjin Lee, Marius Eich, Ming-Hao Liu, Kenji Watanabe, Takashi Taniguchi, Thomas Ihn, and Klaus Ensslin, Transport through a network of topological channels in twisted bilayer graphene, Nano Lett. **18**, 6725 (2018).

[37] Marius Eich, Riccardo Pisoni, Hiske Overweg, Annika Kurzmann, Yongjin Lee, Peter Rickhaus, Thomas Ihn, Klaus Ensslin, František Herman, Manfred Sigrist, Kenji Watanabe, and Takashi Taniguchi, Spin and Valley States in Gate-Defined Bilayer Graphene Quantum Dots, Phys. Rev. X **8**, 031023 (2018).

[38] Marius Eich, Riccardo Pisoni, Alessia Pally, Hiske Overweg, Annika Kurzmann, Yongjin Lee, Peter Rickhaus, Kenji Watanabe, Takashi Taniguchi, Klaus Ensslin, and Thomas Ihn, Coupled quantum dots in bilayer graphene, Nano Lett. **18**, 5042 (2018).

[39] Hiske Overweg, Hannah Eggimann, Xi Chen, Sergey Slizovskiy, Marius Eich, Riccardo Pisoni, Yongjin Lee, Peter Rickhaus, Kenji Watanabe, Takashi Taniguchi, Vladimir Falko, Thomas Ihn, and Klaus Ensslin, Electrostatically induced quantum point contacts in bilayer graphene, Nano Lett. **18**, 553 (2017).

[40] Andrea C. Ferrari *et al.*, Science and technology roadmap for graphene, related two-dimensional crystals, and hybrid systems, Nanoscale **7**, 4598 (2015).

[41] Deep Jariwala, Vinod K. Sangwan, Lincoln J. Lauhon, Tobin J. Marks, and Mark C. Hersam, Emerging device applications for semiconducting two-dimensional transition metal dichalcogenides, ACS Nano **8**, 1102 (2014).

[42] Gianluca Fiori, Francesco Bonaccorso, Giuseppe Iannaccone, Tomás Palacios, Daniel Neumaier, Alan Seabaugh, Sanjay K. Banerjee, and Luigi Colombo, Electronics based on two-dimensional materials, Nat. Nanotechnol. **9**, 768 (2014).

[43] Yuan Liu, Nathan O. Weiss, Xidong Duan, Hung-Chieh Cheng, Yu Huang, and Xiangfeng Duan, Van der Waals heterostructures and devices, Nat. Rev. Mater. **1**, 16042 (2016).

[44] http://github.com/cmt-qo/cmflakes.

[45] Satoru Masubuchi, Eisuke Watanabe, Yuta Seo, Shota Okazaki, Takao Sasagawa, Kenji Watanabe, Takashi Taniguchi and Tomoki Machida, Deep-learning-based image segmentation integrated with optical microscopy for automatically searching for two-dimensional materials, npj 2D Materials and Applications **4**, 3 (2020).

[46] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (O'Reilly Media, Inc., Sebastopol, California, US, 2017).