Deep-learning-based quantum vortex detection in atomic Bose–Einstein condensates

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 035019 https://doi.org/10.1088/2632-2153/abea6a

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
23 December 2020

REVISED
16 February 2021

ACCEPTED FOR PUBLICATION
26 February 2021

PUBLISHED
14 June 2021

PAPER

Deep-learning-based quantum vortex detection in atomic Bose–Einstein condensates

Friederike Metz*, Juan Polo, Natalya Weber and Thomas Busch

Quantum Systems Unit, Okinawa Institute of Science and Technology Graduate University, 1919-1 Tancha, Onna, Okinawa 904-0495, Japan

* Author to whom any correspondence should be addressed.

E-mail: friederike.metz@oist.jp

Keywords: machine learning, object detection, convolutional neural network, vortices, Bose–Einstein condensate, non-equilibrium dynamics, Gross–Pitaevskii equation

Abstract

Quantum vortices naturally emerge in rotating Bose–Einstein condensates (BECs) and, similarly to their classical counterparts, allow the study of a range of interesting out-of-equilibrium phenomena, such as turbulence and chaos. However, the study of such phenomena requires the determination of the precise location of each vortex within a BEC, which becomes challenging when either only the density of the condensate is available or sources of noise are present, as is typically the case in experimental settings. Here, we introduce a machine-learning-based vortex detector motivated by state-of-the-art object detection methods that can accurately locate vortices in simulated BEC density images. Our model allows for robust and real-time detection in noisy and non-equilibrium configurations. Furthermore, the network can distinguish between vortices and anti-vortices if the phase profile of the condensate is also available. We anticipate that our vortex detector will be advantageous for both experimental and theoretical studies of the static and dynamic properties of vortex configurations in BECs.

1. Introduction

In nature, the non-equilibrium behaviour of classical and quantum systems is ubiquitous and includes interesting and complex phenomena, such as turbulence and chaos, which are still only partially understood [1, 2]. Bose–Einstein condensates (BECs) provide a particularly versatile platform for studying and simulating general features of non-equilibrium dynamics, due to the high level of control of experimental systems [3, 4]. In particular, rapidly rotating BECs can support quantum vortices, which are considered to be a key component of superfluid turbulence [5]. Unlike their classical counterparts, quantum vortices are restricted to quantized circulation due to the condition that the wave function has to be single valued at all points. This leads to a well-defined velocity profile that is given by the gradient of the phase [6]. Numerous experiments have generated vortices in BECs [7] and observed the formation of vortex–antivortex pairs [8], vortex rings [9], and vortex lattices [10]. Furthermore, the *in-situ* density imaging of vortex cores has opened the door to the analysis of their real-time dynamics [11–13] and thus, the experimental study of chaos, turbulence, and other out-of-equilibrium dynamics [14–18]. For example, recent results include the detection of persistent vortex clusters emerging from the turbulent flow of high-energy vortex configurations [19, 20], the experimental realization of the quantum analogue of the Kármán vortex street [21], and the observation of vortex–antivortex pairing in a turbulent BEC [22].

However, the study of quantized vortices, and specifically their dynamics, requires their precise location to first be inferred within a BEC [23]. For ground states, the task of detecting vortices is straightforward since they are arranged in a clear pattern with pronounced density minima at their core centers [10, 24] and therefore can be easily spotted by eye or via automated processes. On the other hand, in non-equilibrium configurations, vortices are located at random positions following no distinct order. Furthermore, local density minima that do not correspond to vortices can emerge as a consequence of phononic excitation, making the detection of vortices considerably more difficult [25]. In numerical simulations that model the

dynamics of BECs, one usually has access to the full condensate wave function and hence also to its phase. The phase profile provides a clear indication of the existence of a vortex through a phase winding of 2π around the position of the vortex core. Therefore, vortex detection algorithms for non-equilibrium configurations mainly rely on the BEC phase profile to distinguish vortices from other defects [25, 26]. However, in experiments, the phase profile and thus the information encoded therein is not easily accessible. Moreover, non-zero temperatures and the presence of noise pose an additional challenge to the accurate detection of vortices, and hence require the development of more elaborate methods.

In this paper, we show that a machine-learning-based vortex detector can reliably and accurately locate vortices within out-of-equilibrium BEC density images. It can distinguish vortices from other local density minima, even in situations that are difficult for the human eye. In contrast to conventional vortex-detection algorithms, such as blob detection, the neural network does not require hard-coded features or the fine-tuning of parameters [23, 26]. In addition, the model is robust, i.e. it performs well on simulated data with experimentally relevant sources of noise and generalizes to configurations it has not been trained on, which would not be possible with more traditional object detection methods such as template matching [27]. Hence, we anticipate that our vortex detector can be broadly employed in experimental studies of non-equilibrium vortex configurations where only the BEC density is accessible. On the other hand, in numerical simulations of the BEC, the phase profile is available, and can be provided to the neural network as additional information. In this case, the model can also accurately classify the circulation direction of each vortex.

In recent years, machine learning techniques have become a widely adopted tool in the field of quantum physics [28, 29]. Specifically, in the area of BECs, machine learning methods have been used to optimize the cooling process for the atomic gas [30], learn the Kosterlitz–Thouless transition [31], and devise control schemes for the creation of quantum vortices [32]. On the other hand, deep-learning-based object detection has celebrated remarkable successes in the field of classical computer vision, achieving state-of-the-art results in areas such face, vehicle, and medical image detection [33, 34]. Hence, neural-network-based object detection promises to be a powerful tool for the physical sciences as well, and has already been successfully employed in a few cases [35, 36], for example, to detect and identify characteristics of atomic clouds [37] or to locate dark solitons in a BEC [38]. Finally, let us note that deep learning approaches have also been applied to the detection of vortices in classical fluids, such as locating rotor blade tip vortices [39] or eddies in ocean currents [40]. Motivated by these recent successes, in this work we employ a convolutional neural network (CNN) ansatz for the task of vortex detection which can achieve high accuracies on our test data and is therefore very well suited to the problem of locating vortices in BECs.

This paper is organized as follows: we first present the theoretical model used to simulate BECs and describe how vortices emerge in section 2. Section 3 introduces a machine-learning-based vortex detector. The results of training the model on BEC density images alone are discussed in section 4 and the case of training with density and phase snapshots is presented in section 5. In the appendices, we provide further details and discussion regarding the training data, the network architecture, the evaluation metrics, the features learned by the CNN, and the ability of the network to generalize to different trapping geometries and different levels of noise.

2. Physical system

We consider a dilute and weakly interacting Bose–Einstein condensate rotating around its z-axis with a rotational frequency of Ω. At a temperature of zero, and assuming a tight harmonic confinement in the z direction such that transverse dynamics is frozen out, i.e. $\omega_z \gg \omega_\perp$, we can describe the dynamics of the Bose gas in the co-rotating frame by means of the two-dimensional mean field Gross–Pitaevskii equation (GPE) of the form [41, 42]:

$$i\hbar\frac{\partial}{\partial t}\Psi = \left(-\frac{\hbar^2}{2m}\nabla^2 + \frac{1}{2}m\omega_\perp^2 r^2 + g|\Psi|^2 - \Omega L_z\right)\Psi, \tag{1}$$

where Ψ is the the condensate wave function, $\omega_\perp$ is the frequency of the harmonic trap, $L_z = xp_y - yp_x$ is the angular momentum operator, and $r = \sqrt{x^2+y^2}$ is the radial distance. The effective two-dimensional interaction strength is given by $g = g_{3D}/(\sqrt{2\pi}a_z)$, where $a_z = \sqrt{\hbar/m\omega_z}$ and $g_{3D} = 4\pi\hbar^2 a_s/m$ are the harmonic oscillator length scale of the transverse tight confinement and the three-dimensional interatomic interaction strength, respectively. Here, a_s is the s-wave scattering length. Note that equation (1) is analogous to the more general non-linear Schrödinger equation that can describe a variety of different systems [43]. From here onward, we use harmonic oscillator units by setting $\hbar = \omega_\perp = m = 1$ and choosing interaction strengths $g \in [50, 600]$ as well as rotational frequencies $\Omega \in [0.65, 0.95]$, which correspond to experimentally accessible parameter regimes.

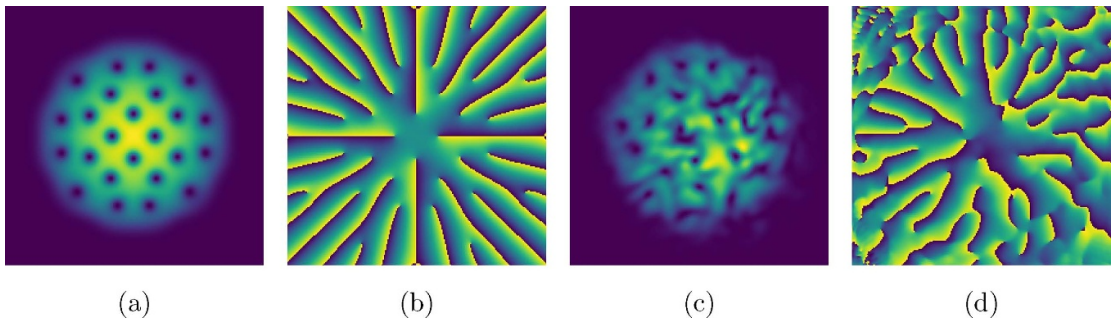

Figure 1. Examples of BEC density and phase profiles for the stationary ground state (a), (b) and for a non-equilibrium configuration (c), (d). The ground state is computed via imaginary time evolution using the GPE with an interaction strength $g = 452$ and a rotational frequency $\Omega = 0.816$. The phase imprinting of additional vortices and a subsequent real-time evolution gives rise to the out-of-equilibrium configuration.

Above a critical rotational frequency Ω_c, the ground state of equation (1) possesses vortices [44–46]. For large rotational frequencies, these vortices arrange themselves into a triangular lattice geometry [10], while for smaller frequencies different configurations can arise [24]. As an example, figures 1(a) and (b) show the numerically obtained density distribution $|\Psi(\mathbf{r})|^2$ and phase profile of the ground-state wave function when $g = 452$ and $\Omega = 0.816$. The vortices are clearly defined through a density dip in their cores and through the characteristic 2π phase winding. Note that the detailed structure of the vortex core depends on the trapping potential [4, 6]: in a homogeneous BEC, the width of the vortex core is fixed by the balance between the kinetic and interaction energies, and a typical core size is given by the healing length $\xi = \sqrt{8\pi n a_s}$, where n corresponds to the density. In trapped systems, the size of the vortex core depends also on the local chemical potential, which gives rise to slightly larger sizes in low-density regions. In addition, vortices surrounded by very low densities in the outer part of the BEC do not contribute to the rotational energy of the system and are therefore irrelevant from a physical point of view [47].

The vortices carried by the ground state all rotate in the same direction, i.e. have a winding number with the same sign, which is determined by the rotational frequency Ω. Situations where vortices with different rotation directions co-exist can be created, for instance, by forcing the superfluid to flow around an obstacle potential [48, 49] or through the process of phase imprinting [50–54]. In the latter case, a single vortex centered at (x_0, y_0) can be generated by applying a phase mask $\phi_{\mathrm{IMP}}(\mathbf{r}) = \arctan(y - y_0, x - x_0)$ with a 2π phase winding in the desired direction. The time evolution of configurations with multiple vortices of unequal rotational directions features interesting out-of-equilibrium processes, such as vortex– antivortex annihilation and the emergence of other low-energy excitations. Furthermore, it has been shown that three and four vortex systems with a single counter-rotating vortex can already lead to chaotic dynamics [14, 51, 55] and that large vortex systems can give rise to quantum turbulence [5, 19, 56, 57]. Figures 1(c) and (d) display a density and phase profile snapshot during a representative time evolution after the phase imprinting of additional anti-vortices. While vortex cores are still clearly visible in the image of the condensate phase, it is more challenging to pinpoint their exact location in the density snapshot.

3. Machine learning model

In the following, we introduce our neural-network-based vortex detector, which is motivated by state-of-the-art object detectors, such as YOLO and Objects as Points [58, 59]. The general task of object detection is to locate each object in an image, draw the corresponding bounding boxes, and associate them to a specific class. Here, we are only interested in detecting vortices, and therefore, our problem reduces to that of binary classification. In section 5, we consider the case where the detector also learns to distinguish between vortices and anti-vortices as two separate classes. Furthermore, since the sizes of vortices across the simulated images do not vary significantly, we focus on predicting the position of each vortex core rather than the full bounding boxes. If necessary, the size of a vortex core can be determined by calculating the healing length of the condensate.

The vortex detector takes as its input grayscale images $I \in [0, 1]^{W \times H \times C}$ with equal width and height, $W = H = 256$, and a number of channels $C = 1, 2$ depending on whether the density profile only or both density and phase profiles are provided to the neural network in two separate channels (see figure 2). In principle, the output of the detector can assign a probability to each image pixel corresponding to whether the pixel represents a vortex core or not. However, due to the large dimensions of the input image, we divide it into a $\frac{W}{R} \times \frac{H}{R}$ grid with $R = 4$, such that each 4×4 grid cell is responsible for detecting at most one object.

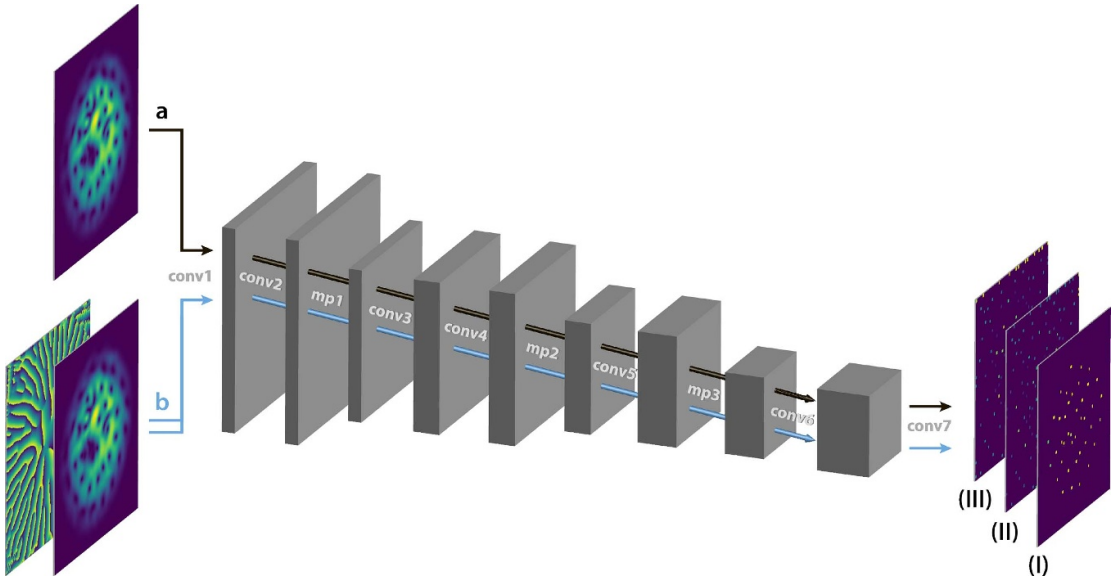

Figure 2. The network takes as its input images of dimension 256 × 256 with the condensate density alone (a), or the density and phase profile as two separate channels (b). The images are fed through seven convolutional and three maxpool layers until the final layer outputs three matrices of dimension 64 × 64. Each entry of the 64 × 64 matrices is associated with a distinct 4 × 4 cell in the original image and represents the probability of a vortex core being present inside the cell (I), and the scaled x (II), and y (III) position of the vortex within that cell.

We estimate the size of vortices in our data set and thus ensure that the grid is chosen sufficiently fine that at most one vortex is present in any cell. The output Y_{ijk} of the neural network is therefore a tensor of dimensions $64 \times 64 \times 3$, where the three channels correspond to the probability of a vortex core being present, and the scaled x and y positions of the core within its grid cell.

In the following, we denote the neural network prediction by Y and the ground-truth label by $\hat{Y}$. The latter are obtained by a brute-force detection method described in detail in appendix A. Our training and test data are comprised of both ground state and out-of-equilibrium configurations, which are obtained through numerical simulations of the GPE (see equation (1)) with parameter values sampled uniformly from the range $g \in [50, 600]$ for the interaction strength and $\Omega \in [0.65, 0.95]$ for the rotational frequency. The obtained density and phase profiles are normalized such that their pixels lie between $[0, 1]$ before being input to the convolutional neural network (CNN). The architecture is composed of seven convolutional layers and three maxpool operations (see figure 2). The full details of the architecture, the training, and the chosen hyperparameters are provided in appendix B. We use the ADAM optimizer [60] and a loss function given by

$$L = \sum_{\text{batch}} \sum_{ij} \Big[- w_1 \hat{Y}_{ij1} \log\left(Y_{ij1}\right) - \left(1 - \hat{Y}_{ij1}\right) \log\left(1 - Y_{ij1}\right) + w_2 \hat{Y}_{ij1} \left(\left(\hat{Y}_{ij2} - Y_{ij2}\right)^2 + \left(\hat{Y}_{ij3} - Y_{ij3}\right)^2 \right) \Big], \tag{2}$$

where w_1, w_2 are hyperparameters. The first term in the loss function is the weighted cross-entropy loss responsible for learning the correct assignment of vortex probabilities for each grid cell. We find that giving a higher weight to learning positive predictions stabilizes training, since otherwise, the network often learns to detect no vortices at all, likely due to the sparsity of vortices within an image. The last term is the mean-squared error (MSE) loss for the x and y positions of the vortex. Note that only those entries of Y with an existing vortex core contribute to this part of the loss function, while all other entries are ignored and in general have arbitrary values. To evaluate and compare the performance of the object detector, we use widely adopted metrics in the field of object detection such as precision, recall, average precision (AP), and the F1 score that we compute for the test data set. For their definitions, we refer the reader to appendix C.

4. Vortex detection using density only

First, we train the object detector directly on density images obtained from simulations with the GPE, i.e. without the addition of any noise. Figures 3(a) and (b) show two representative density images, where white circles correspond to the ground truth and red crosses to the prediction of the trained model. Overall,

Table 1. Detector performance metrics (precision, recall, (mean) average precision (AP), and maximum F1 score) computed on the test data for each trained model (see appendix C for their definitions). In the case of detection using BEC density images only, Gaussian noise is added to the normalized density distributions with mean zero and standard deviations $\sigma = 0.1$ (weak), $\sigma = 0.2$ (moderate), $\sigma = 0.5$ (strong). The stripe pattern was achieved by adding a sinusoidal modulation instead, with amplitudes $A = 0.2$ (weak), $A = 0.5$ (moderate), $A = 1.0$ (strong). Finally, in the case where both the density and the phase profiles are input to the CNN, the Gaussian noise is directly added to the real and imaginary parts of the wave function.

	Precision	Recall	AP	F1
Detection using density only (figure 3)				
(a), (b) w/o noise	96.6	97.2	95.1	96.9
(d) weak Gaussian noise	93.9	93.8	91.1	93.9
(e) moderate Gaussian noise	92.1	90.5	88.2	91.3
(f) strong Gaussian noise	84.7	78.2	78.5	81.3
(g) weak stripes	90.9	90.5	88.2	90.7
(h) moderate stripes	88.4	88.3	83.4	88.4
(i) strong stripes	85.0	83.9	78.8	84.5
Detection using density and phase (figure 4)				
(a), (d) weak Gaussian noise	95.1	95.5	92.4	95.3
(b), (e) moderate Gaussian noise	92.4	92.3	88.0	92.3
(c), (f) strong Gaussian noise	78.0	74.7	69.4	77.2
Detection and classification (figure 5)				
Without noise, vortex–anti-vortex	94.9	96.5	92.3	95.7

we achieve a precision of 96.6% and a recall value of 97.2% on the test data (all other computed evaluation metrics can be inferred from table 1). Precision and recall are calculated through comparison with the ground truth position obtained from the brute-force detection method, which is not always accurate in itself. Hence, our CNN likely performs better than the computed metrics.

While the network detects all vortices in figure 3(a) with nearly perfect accuracy, we observe deviations from the ground truth label in the example shown in figure 3(b). Here, the model detects additional vortices at the boundary of the condensate. However, the corresponding phase profile in figure 3(c) features characteristic phase windings at the locations of the additional detections and, hence, these can be interpreted as vortices as well. In general, we find that in most cases where the number of ground-truth detections and model detections differ, the missing/additional vortices lie at the boundary of the BEC and are often accompanied by a lower confidence probability. Note that the ground-truth labels were obtained using a brute-force detection algorithm which involves applying an arbitrarily chosen mask to the density images, cutting off low-density regions, and therefore excluding any vortices that are not strictly within the BEC (see appendix A for further details). Hence, during training, the neural network also learns that vortices located in very low-density regions should not be detected as such; however, it does not have access to the specific mask used in the brute-force detection algorithm. Therefore, it likely learns a slightly different density cutoff, which gives rise to the additional detections in the test data. In figure D1 of appendix D, we visualize the output of one of the convolutional layers of the trained CNN, which reveals information about the features learned by the model, such as the specific masks applied by the CNN to differentiate regions within and outside the condensate.

To emulate the experimental conditions, we trained separate networks on images with two different sources of noise. The first type is Gaussian random noise with a zero mean, which is added to each pixel of the normalized condensate density images and mimics the measured density distributions in, for example, [61]. We train three independent CNNs, each with a different level of noise, i.e. a different standard deviation ($\sigma = 0.1, 0.2, 0.5$), and plot the resulting predictions together with their ground truths in figures 3(d)–(f). As another example of experimentally relevant noise, we consider stripes in the density images which resemble the fringe patterns that can arise in absorption imaging due to unwanted interference effects [62, 63]. To mimic this pattern, we add a sinusoidal modulation to the density, with a randomly chosen direction and period. In addition, we add Gaussian random noise to the amplitude of the modulation itself. Figures 3(g)–(i) show the corresponding density images together with the model prediction where the amplitude A of the modulation and the amount of noise increases from left to right ($A = 0.2, 0.5, 1.0$, $\sigma = 0.2, 0.5, 1.0$). As expected, for both types of noise considered, the performance of the detector deteriorates as the amount of noise increases, which is also reflected by a smaller precision and recall value (see table 1). In general, we find that, as the noise grows, first only the predicted vortex positions become less accurate, while for larger amounts of noise, the model starts to entirely miss or mistakenly place vortices.

Note that although we have trained separate models for each different level of noise, we observed that each of the trained networks can generalize well to different strengths and types of noise, which is crucial for

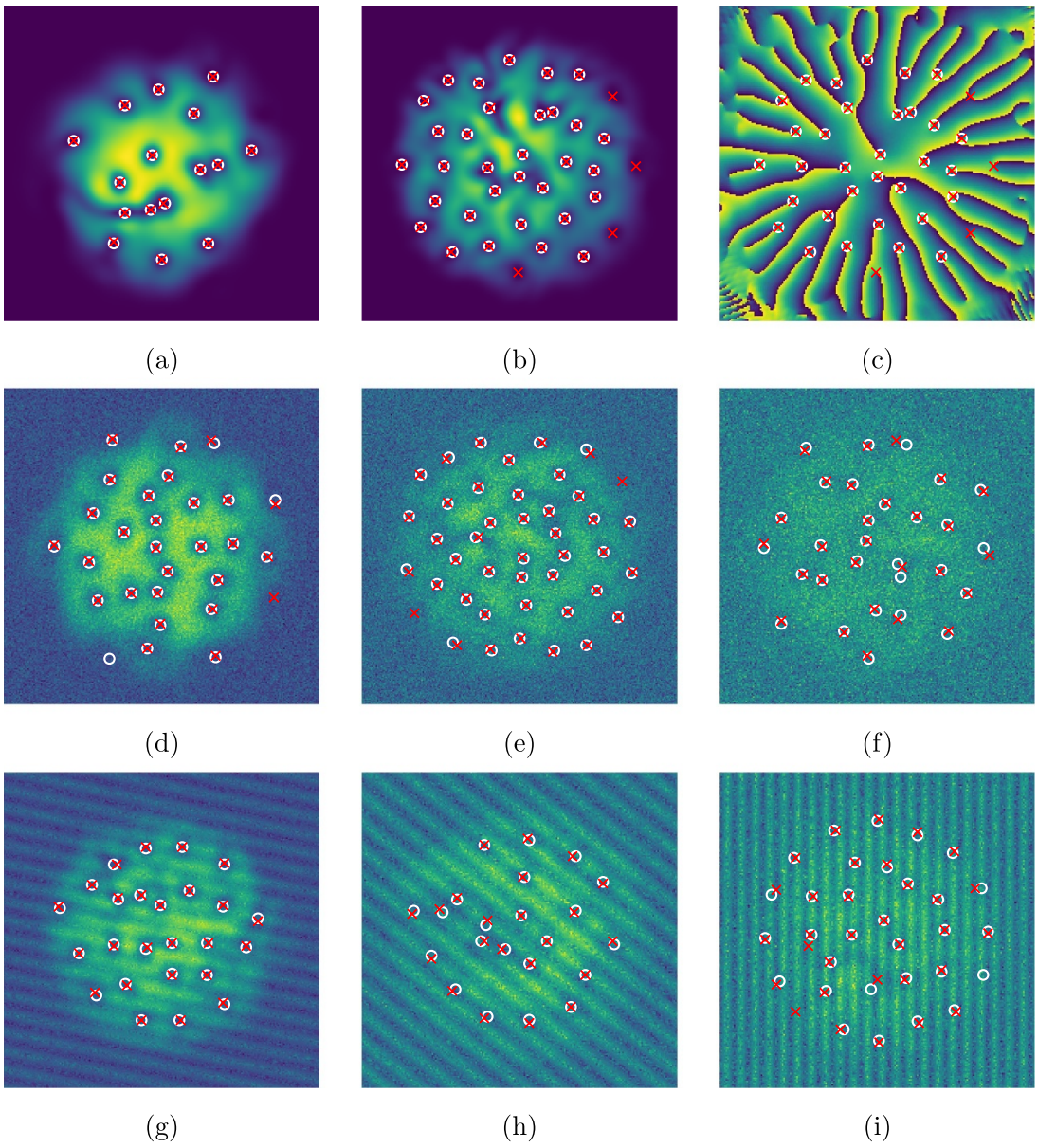

Figure 3. The locations of the vortex cores within each image are indicated by red crosses for the model prediction and by white circles for the ground truth obtained through the brute-force detection method. The CNN model was trained and tested on BEC density images only, and therefore does not have access to the information encoded in the phase profile. (a), (b) Two examples of BEC density configurations; (c) is the corresponding phase profile for the density image in (b), provided here as a guide for the eye. (d)–(f) Density distributions to which random Gaussian noise is added with growing standard deviations from left to right ($\sigma = 0.1, 0.2, 0.5$). In (g)–(i), a sinusoidal modulation with Gaussian noise is added instead, where the amplitude A and the amount of noise increase from left to right ($A = 0.2, 0.5, 1.0$). Note that the pixels in the density images are normalized to lie between $[0, 1]$ before including any noise and before being fed to the neural network.

real experimental situations where the amount of noise will, most likely, change between measured images and experimental runs. For example, the model trained solely on strong Gaussian noise with $\sigma = 0.5$ achieved both a precision and recall of approximately 90% on the test images containing a lower amount of noise and hence performs only slightly worse than the networks that have been directly trained on those data sets. The same model also performs well on images with weak stripes; however, for the case of moderate and strong stripes, the model's performance is considerably worse. This trend is, however, expected since the stripe pattern contains unique features that the model has not been exposed to during training. We also find that a network trained at a lower noise level generalizes to a certain extent to data involving more noise. For instance, the network trained on images with weak stripes achieves good accuracies on test images with weak to moderate Gaussian and stripe noise with $F1$ scores over 80%, and only has difficulties locating vortices in images with a huge amount of noise. A summary of the computed evaluation metrics for the two examples discussed here is given in table E1 in appendix E.

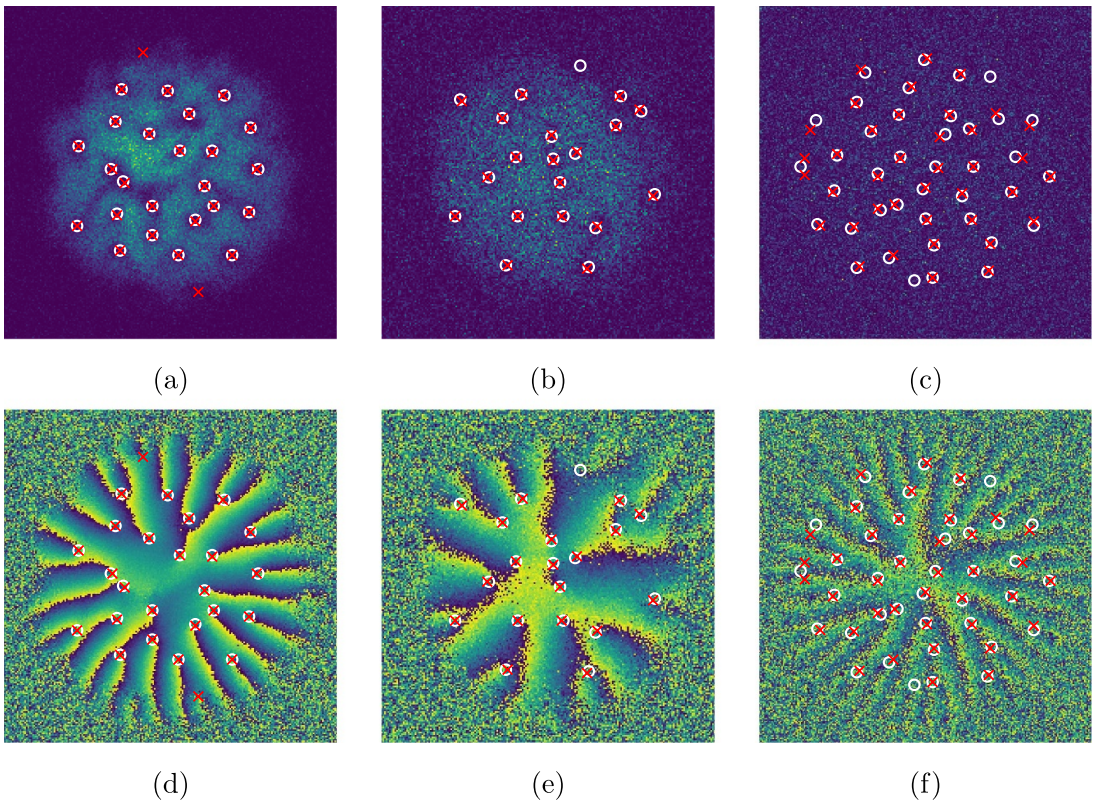

Figure 4. BEC density (a)–(c) and the corresponding phase (d)–(f) configurations after adding Gaussian random noise to the real and imaginary parts of the condensate wave function. The amount of added noise increases from left to right. The model prediction is again denoted by red crosses while the ground truth is indicated by white circles. All images are normalized to lie between $[0, 1]$.

Finally, in appendix F, we show an example in which a model trained solely on images of BECs in a harmonic trap can also detect vortices in BECs in ring-shaped traps without any further training. We therefore expect the model to generalize to similar trapping potentials.

5. Vortex detection using density and phase

In numerical simulations of the GPE, one has access to the full mean-field condensate wave function, rather than just its density. Hence, we can provide both the density and the phase profile as inputs to the CNN in two separate channels in analogy to the three color RGB channels of a conventional image. To make the detection more challenging, we add Gaussian random noise to the real and imaginary parts of the wave function, which gives rise to the density and phase distributions depicted in figures 4(a)–(c) and 4(d)–(f), respectively. We train three models using different levels of noise and show the predicted vortex locations together with their ground truths in figure 4. The achieved performance metrics can be seen in table 1. In particular, for the case of weak noise, the detector performs very well, resulting in precision and recall values that are both above 95%, suggesting that the network exploits the additional information encoded in the BEC phase profile. In figure D1 of appendix D, we display the output of one of the CNN layers, which also indicates that different features are learned depending on whether training is performed on density only, or density and phase images.

Furthermore, access to the phase profile allows us to determine the direction of circulation through the sign of the phase winding. In the images shown in this paper (see, for example, figures 5(d)–(f)) the circulation direction can be easily inferred by checking the direction of the color gradient when moving in a clockwise loop around a vortex core, i.e. whether the color changes continuously from yellow to blue or vice versa. Hence, we next train the network to also classify the sign of circulation for each vortex. The CNN is slightly altered to output four channels; the first two now correspond to the probability of vortices and anti-vortices being present in a specific grid cell, and the last two channels again contain information about the precise location of a detected vortex. The loss function in equation (2) is changed accordingly. Figure 5 shows three exemplary density and phase images, where the model's predictions are represented as crosses and the ground truths as circles, while vortices are depicted in red and anti-vortices are in white. The model

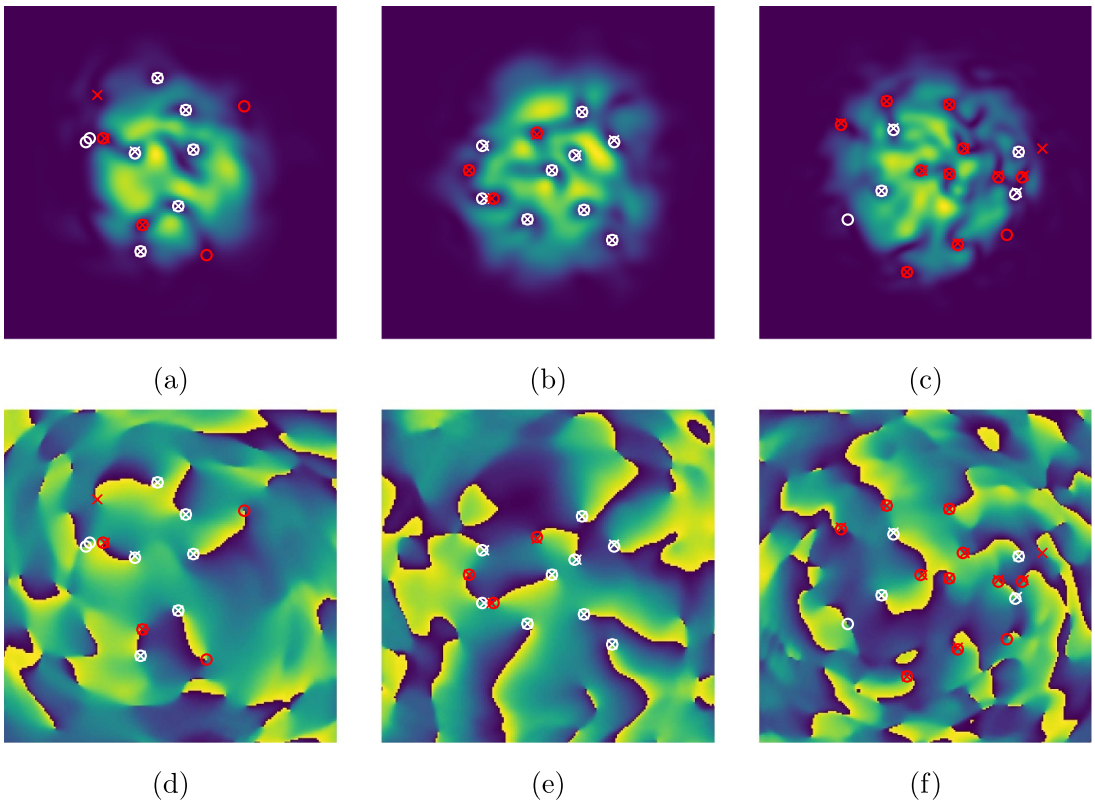

Figure 5. BEC densities (a)–(c) and corresponding phase (d)–(f) configurations. The vortex locations predicted by the network are indicated by crosses and the ground truth by circles. The model was also trained to classify the circulation direction of a vortex; vortices are depicted in red and anti-vortices in white.

can accurately distinguish the circulation direction and in particular, classifies all windings correctly for the images shown here. Moreover, it finds all vortices within the high-density region of the condensate, which is also reflected by the high precision and recall values shown in table 1.

6. Discussion and conclusions

In this work, we have present a machine-learning-based vortex detector that can accurately predict the locations of vortices within two-dimensional BECs trapped in harmonic potentials. The machine-learning model is based on a convolutional neural network (CNN) and takes as its input either an image of the BEC density only, or both the BEC density and phase profiles. We first study the experimentally more relevant case, where only the condensate density is available and which is thus used for training and testing. Without any noise sources, the detector can reliably locate all vortices within an image. Moreover, the model performs well on non-equilibrium configurations that involve local density minima that do not correspond to vortices. Hence, it learns to distinguish density minima arising due to vortices from those caused by other low-energy excitations, even in cases that are challenging for the human eye.

To simulate more realistic experimental conditions, we trained the network on density images with two different types of added noise, that is, Gaussian noise and spurious stripe patterns that arise due to unwanted optical interference effects [61–63]. In both cases, the achieved accuracy of the trained detector decreases with the amount of added noise, as expected. However, the overall performance is still impressive, given that it is nearly impossible to locate any vortices by eye in the images that contain a high level of noise. In contrast to the experimental setting, in numerical simulations, both the density and phase profile of a BEC are available and can therefore also be used to distinguish the circulation direction of each vortex. In this case, our detector learns to correctly classify the sign of circulation as well. Finally, the network can also accurately locate vortices in noisy configurations where Gaussian noise is directly added to the wave function itself. Due to the robustness of the detection against noise, it might be promising to train the detector on configurations generated by the stochastic GPE, which models BECs at finite temperatures [26, 64, 65].

We trained and tested the CNN using ground-truth labels obtained through a brute-force detection algorithm, which already provides the positions of vortices to an excellent precision. This raises the question of whether a machine-learning approach is even necessary and advantageous. However, the brute-force

detection algorithm has certain disadvantages: it involves searching for density minima and checking whether the conditions for a vortex, such as a 2π phase winding, are fulfilled, which is neither efficient nor easily parallelizable. Furthermore, the algorithm does not achieve perfect accuracy itself, i.e. it misses vortices or mistakenly places them at times. Additionally, the method relies heavily on the phase profile of the BEC for labeling out-of-equilibrium configurations, where local density minima may arise due to other excitations in the quantum system. However, in experiments, the phase information is not easily accessible and therefore the algorithm cannot be straightforwardly applied in these settings. Finally, our brute-force method only works for simulated images without noise. While it is, in general, possible to improve the algorithm to also detect vortices in images with specific sources of noise, the implementation is considerably more cumbersome. On the other hand, our machine-learning-based detector is robust to various sources of noise in the input data and does not rely on any hand-engineered features.

The network presented here can be trained in less than an hour on a single GPU and does not require elaborate hyperparameter tuning for any of the tasks considered here. Furthermore, the CNN can process several images in parallel and can therefore detect vortices in a large batch of input images quickly, given that the computation is performed on a GPU. For example, processing a batch of 100 images only takes on the order of milliseconds. The machine-learning model can also be straightforwardly integrated with a GPU solver of the Gross–Pitaevskii equation, which would eliminate the need to transfer data between CPU and GPU [66]. As a possible next step, the vortex detector could be combined with a tracking algorithm, enabling the study of the real-time dynamics of vortices in BECs such as those described in [14, 51, 55]. It would also be interesting to compare the performance of our model to different architecture choices and object detection techniques, and we hope that our results will serve as a benchmark for further research into quantum vortex detection methods.

While the neural network has only been trained on images of BECs in a uniform harmonic trap, we find that the same model can detect vortices in ring-shaped traps without any additional training, and hence, we expect that the detector also generalizes to other trapping geometries of similar symmetry. Moreover, we observe that a model trained on a particular strength and type of noise also works well at different levels of noise. The promising generalization capabilities and the fact that the model performs well on density images alone with sources of noise and in the presence of spurious density minima suggests that the detector will be advantageous for experiments studying the dynamics of vortices in non-equilibrium states [18–20]. Testing the trained model on real experimental data is therefore an interesting future direction of our work.

Data availability statement

The trained models from the main text as well as the code used for training and evaluating the network are available at [67]. The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

This work was supported by OIST Graduate University and we are grateful for the help and support provided by the Scientific Computing and Data Analysis section of the Research Support Division at OIST. JP acknowledges funding from the JSPS KAKENHI Grant No. 20K14417.

Appendix A. Training data generation

Object detection belongs to the category of supervised learning tasks and therefore requires labeled training data. To include a variety of different vortex configurations for training, we use both ground states and non-equilibrium states generated within different parameter regimes. For each training example, an interaction strength g and a rotational frequency Ω are uniformly sampled in the range $g \in [50, 600]$ and $\Omega \in [0.65, 0.95]$. The ground state is obtained through imaginary time evolution using the split-step method [68]. To create out-of-equilibrium configurations containing both vortices and anti-vortices, we employ the method of phase imprinting [53], where between four and seven vortices are placed at random locations and with random circulation directions. We perform a short imaginary time propagation that simulates a small thermal relaxation of the system and a subsequent real-time evolution, after which, a snapshot of the condensate wave function is saved. The extracted condensate density and phase images comprise the training and test input data, which include 1000 ground state samples and another 1000 out-of-equilibrium samples. The combined data set of 2000 images is randomly split into a training set containing 1600 images and a test set including the remaining 400 images.

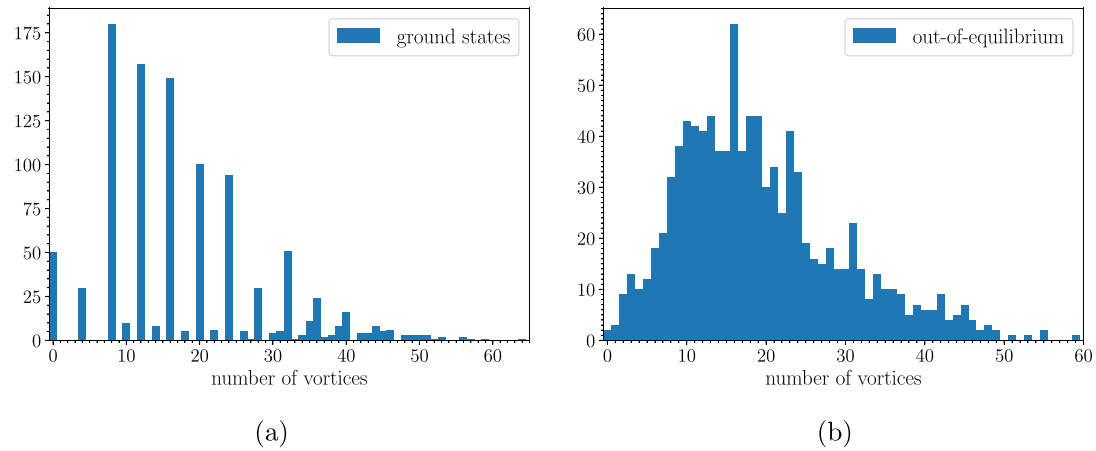

Figure A1. Distribution of the number of vortices across the training set for ground states (a) and out-of-equilibrium configurations (b). The combined distribution has a mean value of 18.7 vortices per image.

The ground truth labels for each image are the positions of all vortex cores inside the BEC. We obtain the x and y coordinates within pixel resolution through a combination of different techniques. We first apply a mask to the density and phase profiles, cutting off regions outside the BEC. Hence, we ensure that only vortices strictly within the condensate are detected. As the cutoff threshold, we choose 15% of the maximum density $|\psi|^2$. Next, we find all local density minima within an image. For each local minimum, we calculate the phase gradient along a closed loop centered at the minimum, check whether the slope equals ± 1, and that the loop adds up to $\pm 2\pi$, thus displaying the characteristic 2π phase winding. If all these conditions are met, the corresponding pixel position is stored in the list of labels. Note that this brute-force method of detecting vortices is not perfectly accurate and misses or mistakenly places vortices in a few cases. Hence, some of the labels used for training are corrupt, however, the overall excellent performance of the detector on the test data suggests that the training of the network is robust against the errors in the training set.

We found that the number of vortices within a single image varied between 0 and 65 in our data set. The distribution of the number of vortices across all images is shown in figure A1. The dependence of the number of vortices on the applied rotation frequency Ω is non-linear, i.e. for a large range of sampled rotational frequencies, the number of vortices increases only slowly, while in the high-frequency regime, the number grows more rapidly [69]. This effect can be observed in the distributions, which are asymmetric and slightly shifted towards a lower number of vortices with a mean value of 18.7 vortices per image.

Appendix B. Neural network architecture and training details

The architecture of the neural-network based vortex detector is based on SlimNet, a convolutional neural network (CNN) specifically designed for detecting small objects [70]. It contains convolutional as well as maxpool layers, as depicted in table B1. The network takes as its input images of size 256×256, which can be either the density profile or the density and phase profiles provided in two separate channels. All convolutional layers are followed by relu activations except for the last layer, which uses a sigmoid activation instead. The output Y of the network after the final convolutional layer is a $64 \times 64 \times 3$ tensor, where the three channels correspond to the probability of detection and the scaled x and y positions, respectively. For example, an output $Y_{ij1} = 1$ would indicate that a vortex is present in the grid cell denoted by ij and the precise position of the core within that grid cell can be read off by checking Y_{ij2} for the x and Y_{ij3} for the y coordinates. On the other hand, all grid cells where $Y_{ij1} = 0$ do not contain a vortex, and therefore the second and third output channels can be ignored in such cases. In the case where the circulation direction of a given vortex is also classified, the output contains four channels, where the first two represent the probability of a vortex and anti-vortex, respectively. The final maxpool layer serves as a non-max suppression to eliminate multiple detections of the same vortex.

We implement the CNN and train it using the machine learning library Flux, available in Julia [71]. We use the ADAM optimizer [60] with a batch size of 100, a learning rate $\eta = 0.001$, and decay rates of $\beta_1 = 0.9$ for the first and $\beta_1 = 0.999$ for the second momentum estimates. The weights in the loss function of equation (2) are set to $w_1 = w_2 = 10$ and the network is trained for 100–500 epochs, depending on the learning task. A hundred epochs of training took on the order of 10 min on a NVIDIA TITAN X Pascal GPU.

Table B1. Neural network architecture. The number of input and output channels (marked by [*]) differs between learning tasks. The final maxpool layer serves as a non-max suppression.

Layer	Filter	Stride	Pad	Channels
conv 1	3×3	1	1×1	$1/2^* \rightarrow 10$
conv 2	3×3	1	1×1	$10 \rightarrow 10$
maxpool 1	2×2	2		$10 \rightarrow 10$
conv 3	3×3	1	1×1	$10 \rightarrow 20$
conv 4	3×3	1	2×2	$20 \rightarrow 20$
maxpool 2	2×2	2		$20 \rightarrow 20$
conv 5	3×3	1	1×1	$20 \rightarrow 30$
maxpool 3	2×2	1		$30 \rightarrow 30$
conv 6	3×3	1	1×1	$30 \rightarrow 40$
conv 7	1×1	1		$40 \rightarrow 3/4^*$
maxpool 4	3×3	1	1×1	$3/4^* \rightarrow 3/4^*$

A subsequent forward pass took 0.0025 s for a single image and 0.009 s for a batch of 100 images, therefore allowing for real-time detection once the model is successfully trained.

Appendix C. Evaluation metrics

To evaluate and compare the performance of the vortex detector for each different learning task, we use standard metrics in the field of object detection such as precision and recall [33]. Precision describes how many of the detections within an image are accurate, while recall quantifies how many of the actual objects in an image are detected. Denoting true positives by *TP*, false positives by *FP*, and false negatives by *FN*, precision and recall are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{C.1}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{C.2}$$

The machine learning model outputs the probability that a vortex is present in each grid cell. A confidence threshold is used to discard low-probability detections and label high-confidence predictions as positives, *P*. Conventionally, a larger confidence threshold increases precision while decreasing recall, and vice versa. For each different detection task, we calculate an optimal confidence threshold that maximizes the harmonic mean of precision and recall, which we describe further below. To distinguish true positives from false positives, the object detection community usually computes the intersection over the union (IoU) given by the ratio of the intersection area and the union area of the detected bounding box and the ground-truth bounding box. If the IoU is larger than a predefined value, the detection is considered to be a true positive *TP*, whereas it is labeled as a false positive *FP* otherwise. In our case, the vortices across images are of similar sizes and therefore we can use a simple distance measure between the detected vortex position and the ground-truth position instead of the IoU. We choose the pixel-wise Euclidean distance and an arbitrary distance threshold of $\sqrt{5}$. Hence, all detections that are within $\sim$2 pixels of their ground-truth position are identified as positives.

As mentioned before, there is a trade-off between precision and recall which is controlled by the value of the confidence score threshold. To examine the performance of the model across different confidence thresholds, we plot precision against recall for ten different threshold values in figure C1. Each curve of a different color corresponds to one of the separately trained models considered in the main text. From each point on a curve, we can determine the F1 score, i.e. the harmonic mean between precision and recall:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{C.3}$$

The optimal confidence threshold corresponds to the maximum F1 score, which is used for generating all labeled plots within this paper. Furthermore, we calculate the average precision (AP) as the mean over the precision values $p(r)$. Finally, we also provide the precision and recall values computed at the optimal confidence threshold as another meaningful performance metric of the vortex detector. In the case of the multi-class detection considered in section 5, the AP and F1 scores are first calculated separately for each class and then averaged to obtain the mean average precision and the mean F1 score.

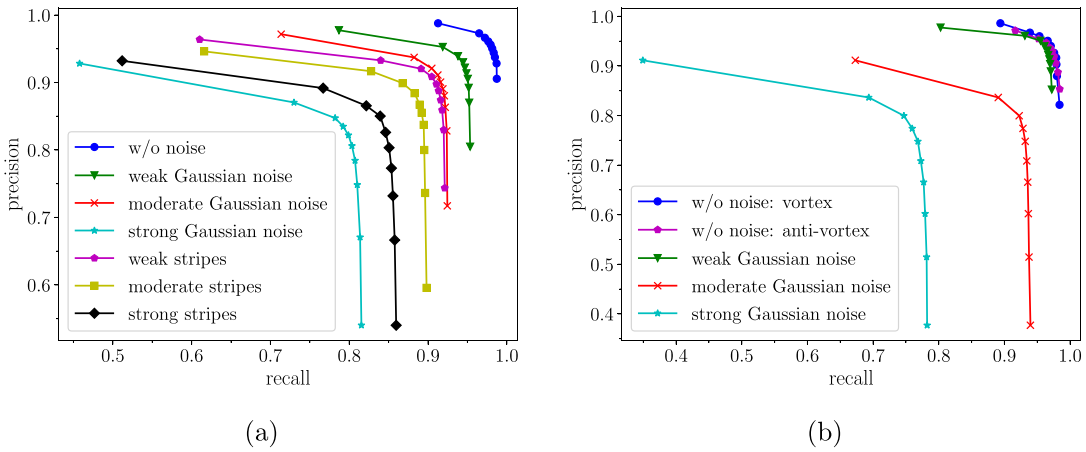

Figure C1. Precision–recall curves for the training tasks of section 4(a) and section 5(b). Precision/recall values are calculated for ten different confidence thresholds in the equally spaced interval between [0.05, 0.95]. The precision/recall values corresponding to the highest F1 score are shown in table 1 of the main text, together with the maximum F1 score, and the average of the precision values (AP).

Appendix D. Visualization of the CNN layers

To elucidate the inner workings of the trained CNN, we visualize the output after the fourth convolutional layer for a particular input image in figure D1. Each image corresponds to the output of one of the 20 channels and gives information about the features learned by the network. Figure D1(a) shows the feature maps after training using BEC density images only. Here, the network clearly learns to separate the condensate from the background by applying different masks. It also detects all density minima within the condensate; however, each channel seems to focus on slightly different characteristics such as the depth, size, or shape of a minima. On the other hand, the feature maps plotted in figure D1(b) were obtained when training with phase and density images. In this case, an interpretation is less evident, but we can observe that the network takes advantage of the additional information supplied by the phase profile. Interestingly, we found that the output of CNN layers trained on noisy images did not differ significantly from the ones shown here. Therefore, the model also learns to de-noise the input if necessary, and thus ignores any spurious features contained in the noise itself.

Appendix E. Generalization to different sources and levels of noise

The models considered in the main text of this paper were all trained on data sets containing a specific source and level of noise. However, real experimental images will not be subject to just a single source of noise and the amount of noise will, most likely, vary between images. Therefore, we also test how well a model trained on a specific noise configuration generalizes to other types and levels of noise. As two examples, we consider a model trained only on data with strong Gaussian noise, or only on images with weak stripes. We show the computed performance metrics for both models tested on all the different data sets in table E1. The values suggest that the networks indeed generalize to unseen noise strengths and types, especially if the amount of noise the model is tested on is lower compared to the one present in the training data.

Appendix F. Generalization to different trap geometries

The networks considered in this paper have been trained solely on BEC configurations in a harmonic trap. To examine whether the same model can be used to detect vortices in different trap geometries, we generate additional images of a BEC in a ring-shaped trap. The potential in the GPE (equation (1)) is replaced by $V = \frac{1}{2}m\omega_r^2(r - r_0)^2$ where r_0 is the toroidal radius and ω_r is the radial trapping frequency. An example of the resulting condensate density and phase profile after real time evolution is shown in figures F1(a) and (d). We test some of our previously trained networks on these new configurations without any further training and indicate the predictions together with their ground truth in figure F1. The model is able to accurately locate

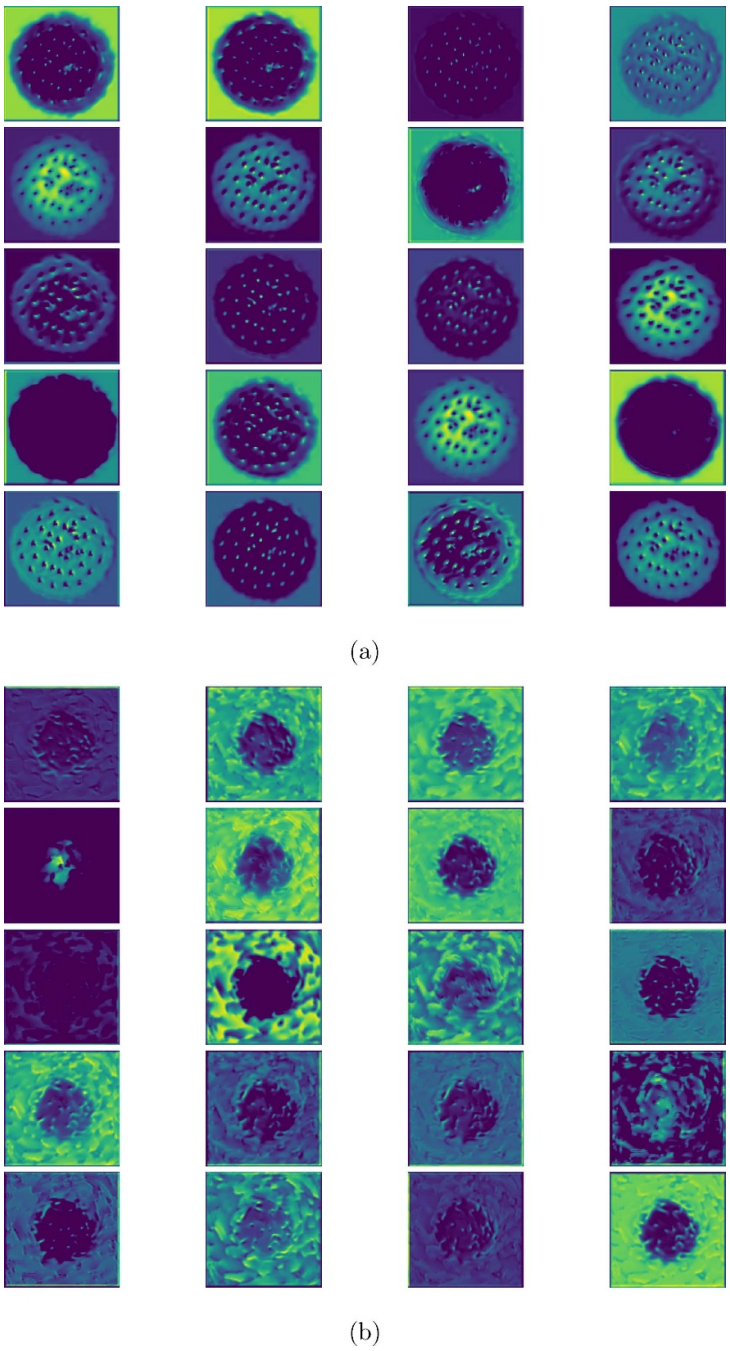

Figure D1. Example output after the fourth convolutional layer of the CNN when trained on density images only (a) or on density and phase images (b).

the vortices within the high-density regions for images without (figure F1(a)) and with noise (figures F1(b) and (c)), while it detects additional vortices at the border of the condensate. Increasing the confidence threshold will likely result in fewer false positive detections. Due to the overall good performance that we observe for the ring-shaped potential, we expect that our trained models will generalize well to other trap geometries of similar symmetry.

Table E1. Detector performance metrics (precision, recall, average precision (AP), and maximum F1 score) for a model trained on images with (a) strong Gaussian noise ($\sigma = 0.5$) and (b) weak stripe noise ($A = 0.1$). Each of the two models is tested on all data sets containing different strengths and/or sources of noise. The type of noise marked by [*] represents the corresponding data set on which the model was trained. The Gaussian noise is added to the normalized BEC density with mean zero and standard deviations $\sigma = 0.1$ (weak), $\sigma = 0.2$ (moderate), and $\sigma = 0.5$ (strong). The stripe pattern results from adding sinusoidal modulations to the normalized density with amplitudes $A = 0.2$ (weak), $A = 0.5$ (moderate), and $A = 1.0$ (strong).

	Precision	Recall	AP	F1
(a) Model trained on data with strong Gaussian noise and tested on data with:				
No noise	91.2	91.4	82.1	91.3
Weak Gaussian noise	92.3	89.3	83.3	90.8
Moderate Gaussian noise	91.0	87.7	83.3	89.3
Strong Gaussian noise*	84.7	78.2	78.5	81.3
Weak stripes	87.6	81.3	71.9	84.3
Moderate stripes	65.9	58.4	51.0	61.9
Strong stripes	46.4	44.8	37.8	45.6
(b) Model trained on data with weak stripes and tested on data with:				
No noise	91.6	91.6	71.1	91.6
Weak Gaussian noise	90.1	82.8	86.6	86.3
Moderate Gaussian noise	83.1	76.1	85.8	79.5
Strong Gaussian noise	61.6	41.0	64.2	49.3
Weak stripes*	90.9	90.5	88.2	90.7
Moderate stripes	83.4	77.1	79.4	80.1
Strong stripes	61.7	53.1	58.3	57.1

* The type of noise marked by a star represents the corresponding data set on which the model was trained.

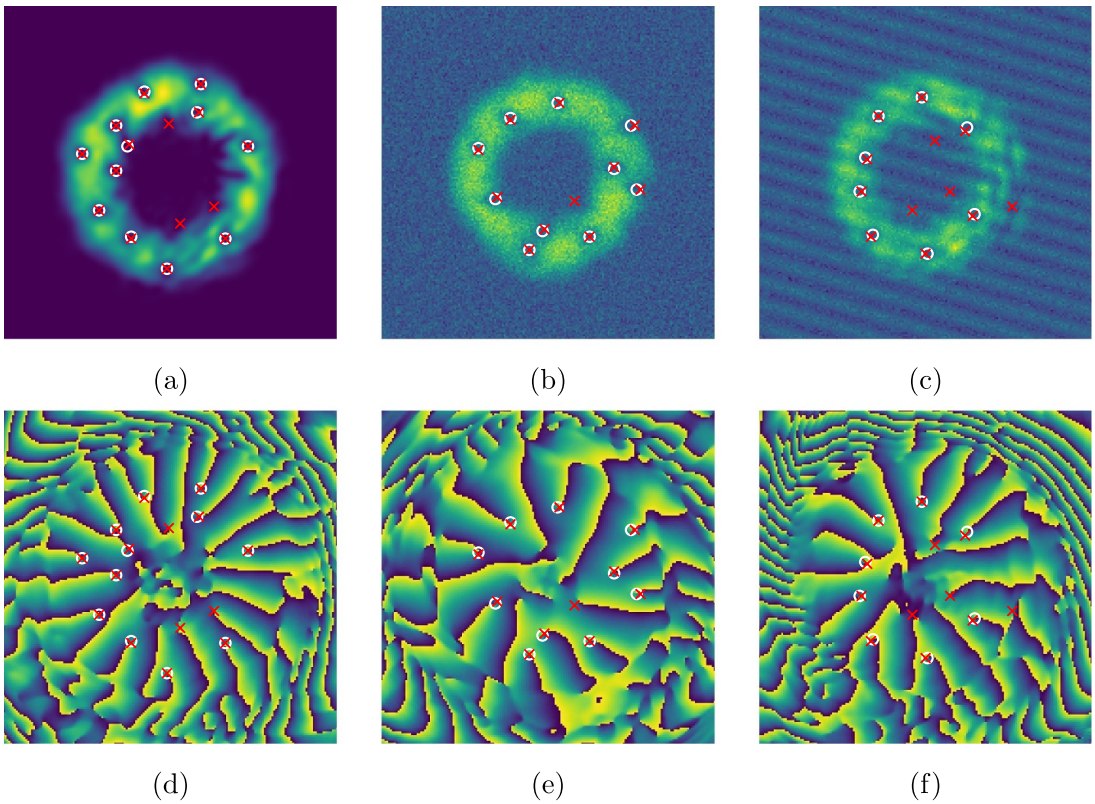

Figure F1. Density (a)–(c) and corresponding phase (d)–(f) configurations for a BEC in a ring-shaped trap. Weak Gaussian noise is added to the density in (b) and a stripe pattern is added to the density in (c). Each density image is fed through the corresponding trained network as described in the main text, with the resulting predictions displayed as red crosses and the ground truth as white circles. The phase profile is shown for pure visualization purposes; that is, only the density images were used as input to the CNN.

ORCID iDs

Friederike Metz https://orcid.org/0000-0002-4745-7329
Juan Polo https://orcid.org/0000-0003-0742-6015
Natalya Weber https://orcid.org/0000-0002-1955-3612
Thomas Busch https://orcid.org/0000-0003-0535-2833

References

[1] Dorfman J R 1999 *An Introduction to Chaos in Nonequilibrium Statistical Mechanics* (*Cambridge Lecture Notes in Physics*) (Cambridge: Cambridge University Press)
[2] Špička V, Keefe P D and Nieuwenhuizen T M 2019 *Eur. Phys. J. Spec. Top.* **227** 1837–48
[3] Dalfovo F, Giorgini S, Pitaevskii L P and Stringari S 1999 *Rev. Mod. Phys.* **71** 463–512
[4] Fetter A L and Svidzinsky A A 2001 *J. Phys.: Condens. Matter* **13** R135–94
[5] White A C, Anderson B P and Bagnato V S 2014 *Proc. Natl Acad. Sci.* **111** 4719–26
[6] Parker N G, Jackson B, Martin A M and Adams C S 2008 *Vortices in Bose–Einstein Condensates: Theory* (Berlin: Springer) pp 173–89
[7] Chevy F 2008 *Vortices in Bose–Einstein Condensates: Experiments* (Berlin: Springer) pp 191–207
[8] Inouye S, Gupta S, Rosenband T, Chikkatur A P, Görlitz A, Gustavson T L, Leanhardt A E, Pritchard D E and Ketterle W 2001 *Phys. Rev. Lett.* **87** 080402
[9] Anderson B P, Haljan P C, Regal C A, Feder D L, Collins L A, Clark C W and Cornell E A 2001 *Phys. Rev. Lett.* **86** 2926–9
[10] Abo-Shaeer J R, Raman C, Vogels J M and Ketterle W 2001 *Science* **292** 476–9
[11] Freilich D V, Bianchi D M, Kaufman A M, Langin T K and Hall D S 2010 *Science* **329** 1182–5
[12] Wilson K E, Newman Z L, Lowney J D and Anderson B P 2015 *Phys. Rev.* A **91** 023621
[13] Haljan P C, Coddington I, Engels P and Cornell E A 2001 *Phys. Rev. Lett.* **87** 210403
[14] Navarro R, Carretero-González R, Torres P J, Kevrekidis P G, Frantzeskakis D J, Ray M W, Altuntaş E and Hall D S 2013 *Phys. Rev. Lett.* **110** 225301
[15] Serafini S, Galantucci L, Iseni E, Bienaimé T, Bisset R N, Barenghi C F, Dalfovo F, Lamporesi G and Ferrari G 2017 *Phys. Rev.* X **7** 021031
[16] Kwon W J, Moon G, Choi J Y, Seo S W and Shin Y I 2014 *Phys. Rev.* A **90** 063627
[17] Neely T W *et al* 2013 *Phys. Rev. Lett.* **111** 235301
[18] Reeves M T, Goddard-Lee K, Gauthier G, Stockdale O R, Salman H, Edmonds T, Yu X, Bradley A S, Baker M, Rubinsztein-Dunlop H, Davis M J and Neely T W 2020 arXiv:2010.10049
[19] Johnstone S P, Groszek A J, Starkey P T, Billington C J, Simula T P and Helmerson K 2019 *Science* **364** 1267–71
[20] Gauthier G, Reeves M T, Yu X, Bradley A S, Baker M A, Bell T A, Rubinsztein-Dunlop H, Davis M J and Neely T W 2019 *Science* **364** 1264–7
[21] Kwon W J, Kim J H, Seo S W and Shin Y 2016 *Phys. Rev. Lett.* **117** 245301
[22] Seo S W, Ko B, Kim J H and Shin Y 2017 *Sci. Rep.* **7** 4587
[23] Rakonjac A, Marchant A L, Billam T P, Helm J L, Yu M M H, Gardiner S A and Cornish S L 2016 *Phys. Rev.* A **93** 013607
[24] Aftalion A and Du Q 2001 *Phys. Rev.* A **64** 063603
[25] Ortega A B, Bucio-Pacheco S, Lopez-Huidobro S, Perez-Garcia L, Poveda-Cuevas F J, Seman J A, Arzola A V and Volke-Sepúlveda K 2019 *Opt. Express* **27** 4105–15
[26] Groszek A J, Davis M J and Simula T P 2020 *SciPost Phys.* **8** 39
[27] Brunelli R 2009 *Template Matching Techniques in Computer Vision: Theory and Practice* (New York: Wiley)
[28] Dunjko V and Briegel H J 2018 *Rep. Prog. Phys.* **81** 074001
[29] Carrasquilla J 2020 *Adv. Phys.* X **5** 1797528
[30] Barker A J, Style H, Luksch K, Sunami S, Garrick D, Hill F, Foot C J and Bentine E 2020 *Mach. Learn.: Sci. Technol.* **1** 015007
[31] Beach M J S, Golubeva A and Melko R G 2018 *Phys. Rev.* B **97** 045207
[32] Saito H 2020 *J. Phys. Soc. Japan* **89** 074006
[33] Xiao Y, Tian Z, Yu J, Zhang Y, Liu S, Du S and Lan X 2020 *Multimed. Tools. Appl.* **79** 23729–91
[34] Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X and Pietikäinen M 2020 *Int. J. Comput. Vis.* **128** 261–318
[35] Minor E N, Howard S D, Green A A S, Glaser M A, Park C S and Clark N A 2020 *Soft Matter* **16** 1751–9
[36] Usman M, Wong Y Z, Hill C D and Hollenberg L C L 2020 *NPJ Comput. Mater.* **6** 19
[37] Hofer L R, Krstajić M, Juhász P, Marchant A L and Smith R P 2020 arXiv:2011.10536
[38] Guo S, Fritsch A R, Greenberg C, Spielman I B and Zwolak J P 2021 arXiv:2101.05404
[39] Luo Y, Shao Y, Chu H, Wu B, Huang M and Rao Y 2020 CNN-based blade tip vortex region detection in flow field *11th Int. Conf. Graphics and Image Processing* vol 11373 (SPIE) pp 182–7
[40] Bai X, Wang C and Li C 2019 *IEEE Access* **7** 106336–45
[41] Salasnich L, Parola A and Reatto L 2002 *Phys. Rev.* A **65** 043614
[42] Petrov D S, Holzmann M and Shlyapnikov G V 2000 *Phys. Rev. Lett.* **84** 2551–5
[43] Liu W M and Kengne E 2019 *Overview of Nonlinear Schrödinger Equations* (Singapore: Springer) pp 1–13
[44] Lundh E, Pethick C J and Smith H 1997 *Phys. Rev.* A **55** 2126–31
[45] Chevy F, Madison K W and Dalibard J 2000 *Phys. Rev. Lett.* **85** 2223–7
[46] Madison K W, Chevy F, Wohlleben W and Dalibard J 2000 *Phys. Rev. Lett.* **84** 806–9
[47] Tsubota M, Kasamatsu K and Ueda M 2002 *Phys. Rev.* A **65** 023603
[48] Sasaki K, Suzuki N and Saito H 2010 *Phys. Rev. Lett.* **104** 150404
[49] Neely T W, Samson E C, Bradley A S, Davis M J and Anderson B P 2010 *Phys. Rev. Lett.* **104** 160401
[50] O'Riordan L J and Busch T 2016 *Phys. Rev.* A **94** 053603
[51] Zhang T, Schloss J, Thomasen A, O'Riordan L J, Busch T and White A 2019 *Phys. Rev. Fluids* **4** 054701
[52] Leanhardt A E, Görlitz A, Chikkatur A P, Kielpinski D, Shin Y, Pritchard D E and Ketterle W 2002 *Phys. Rev. Lett.* **89** 190403
[53] Dobrek L, Gajda M, Lewenstein M, Sengstock K, Birkl G and Ertmer W 1999 *Phys. Rev.* A **60** R3381–4
[54] Andersen M F, Ryu C, Cladé P, Natarajan V, Vaziri A, Helmerson K and Phillips W D 2006 *Phys. Rev. Lett.* **97** 170406

[55] Koukouloyannis V, Voyatzis G and Kevrekidis P G 2014 *Phys. Rev.* E **89** 042905
[56] Wilson K E, Samson E C, Newman Z L, Neely T W and Anderson B P 2013 *Experimental Methods for Generating Two-Dimensional Quantum Turbulence in Bose–Einstein Condensates* (Singapore: World Scientific) pp 261–98
[57] White A C, Barenghi C F, Proukakis N P, Youd A J and Wacks D H 2010 *Phys. Rev. Lett.* **104** 075301
[58] Redmon J, Divvala S, Girshick R and Farhadi A 2016 You only look once: unified, real-time object detection *Conf. on Computer Vision and Pattern Recognition (CVPR)* pp 779–88
[59] Zhou X, Wang D and Krähenbühl P 2019 Objects as points (arXiv:1904.07850)
[60] Kingma D P and Ba J 2015 Adam: a method for stochastic optimization *3rd Int. Conf. Learning Representations* (*San Diego*) (available at:http://arxiv.org/abs/1412.6980)
[61] Fölling S, Gerbier F, Widera A, Mandel O, Gericke T and Bloch I 2005 *Nature* **434** 481–4
[62] Ness G, Vainbaum A, Shkedrov C, Florshaim Y and Sagi Y 2020 *Phys. Rev. Appl.* **14** 014011
[63] Song B, He C, Ren Z, Zhao E, Lee J and Jo G B 2020 *Phys. Rev. Appl.* **14** 034006
[64] Gautam S, Roy A and Mukerjee S 2014 *Phys. Rev.* A **89** 013612
[65] Jackson B, Proukakis N P, Barenghi C F and Zaremba E 2009 *Phys. Rev.* A **79** 053615
[66] Schloss J R and O'Riordan L J 2018 *J. Open Source Softw.* **3** 1037
[67] Metz F 2020 Deep learning based quantum vortex detection in atomic Bose–Einstein condensates (available at: https://github.com/frmetz/quantum_vortex_detection)
[68] Javanainen J and Ruostekoski J 2006 *J. Phys. A: Math. Gen.* **39** L179–84
[69] Kasamatsu K, Tsubota M and Ueda M 2003 *Phys. Rev.* A **67** 033610
[70] Yang Z, Liu Y, Liu L, Tang X, Xie J and Gao X 2019 *IEEE Trans. Geosci. Remote Sens.* **57** 8445–57
[71] Innes M 2018 *J. Open Source Softw.* **3** 602