Bose-Einstein condensate experiment as a nonlinear block of a machine learning pipeline

Maurus Hans, Elinor Kath,* Marius Sparn, Nikolas Liebster, Helmut Strobel, and Markus K. Oberthaler
Kirchhoff-Institut für Physik, Im Neuenheimer Feld 227, Universität Heidelberg, Heidelberg, Germany

Felix Draxler
Computer Vision and Learning Lab, Universität Heidelberg and Image and Pattern Analysis Group

Christoph Schnörr
Image and Pattern Analysis Group, Universität Heidelberg, Heidelberg, Germany

Physical systems can be used as an information processing substrate and, with that, extend traditional computing architectures. For such an application, the experimental platform must guarantee pristine control of the initial state, the temporal evolution, and readout. All these ingredients are provided by modern experimental realizations of atomic Bose-Einstein condensates. By embedding a quantum gas experiment in a machine learning pipeline, one can represent nonlinear functions while only linear operations on classical computers of the pipeline are necessary. We demonstrate successful regression and interpolation of a nonlinear function using an elongated cloud of potassium atoms and characterize the performance of our system.

I. INTRODUCTION

A very general approach to harness the computational resources of a system is to embed it in a machine learning pipeline, which processes data sequentially. Each block in the pipeline must be able to compute functions of its input. For example, a continuous-time input can be processed by a dynamical system, such as a recurrent artificial neural network [1–4]. Physical systems can also compute expressive functions, making them strong candidates for a computational block in a machine learning pipeline, if they are reliable. This has been demonstrated previously [5–7].

Only recently, it has been suggested that a Bose-Einstein condensate (BEC) of ultracold atoms is an ideal system to be embedded in a machine learning pipeline [8,9], as the complex nonlinear dynamics of the BEC unambiguously depend on the input. Here we show that the advanced control of BEC experiments enables the physical implementation of such schemes, with the experiment as a nonlinear layer of a machine learning (ML) pipeline. We address the concrete task of computing the value $u = f(x)$ of a given nonlinear function f, employing a standard computer for linear operations only. Thus we perform a regression, which is another application of such models besides the common task of classification. In the context of machine learning, this is known as the linear basis function model [10, Chap. 3.1].

---

*physical_computing_1@matterwave.de

II. THE MODEL

The structure of the overall scheme is depicted in Fig. 1(a) and consists of an input block, an experimental block including a Bose-Einstein condensate as the central element, and a linear output block, where the latter is defined by supervised learning [see Fig. 1(b)]. In the first, linear block, one of N_x input values x_i is mapped onto the mean-field degree of freedom of a BEC, namely, the complex order parameter $\Psi = \psi e^{i\phi}$, with amplitude $\psi(z)$ and phase $\phi(z)$. As the representation of the number x_i we choose a step in the phase at a certain spatial position z_i. The experimental block is defined by the imprinting of the phase, a subsequent evolution, and a readout of the density. The resulting one-dimensional density $n_i(z) = |\Psi_i(z)|^2$ corresponds to the input x_i. The linear output block consists of two layers: The first improves signal to noise by binning the density $n_i(z)$ to N_w bins, leading to a data vector $\bar{\mathbf{n}}_{\mathbf{i}} = (n_{i,1}, \ldots, n_{i,N_w})$. The second layer performs a linear transformation $y_i = \bar{\mathbf{n}}_{\mathbf{i}} \cdot \mathbf{w}$, where the weight vector $\mathbf{w} = (w_1, \ldots, w_{N_w})$ is a result of supervised learning. In the example given in Fig. 1(b), the function $f(x) = \sin(\pi x)/(\pi x)$ is implemented, and for the input value of $x_{42} = -0.5152$ we find $y_{42} = 0.6084$, close to $u_{42} = f(-0.5152) = 0.6172$.

III. IMPLEMENTATION

The central part of our experimental system is an elongated BEC of ^{39}K atoms with a length of 85 μm and a width of 6 μm confined in a box trap. The condensate is confined in the gravitational direction with a trap frequency of 1.5 kHz, realized with a blue-detuned (532 nm) lattice with a spacing of ∼5 μm. In the other direction, a digital micromirror device (DMD) is used to realize the elongated configuration, employing laser light at the same wavelength. The phase step encoding the input is experimentally implemented by shining
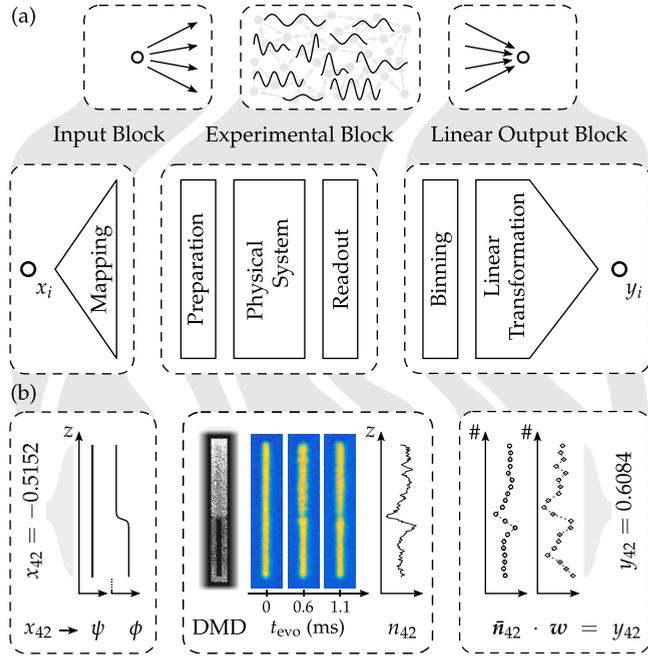
FIG. 1. Machine learning pipeline. (a) In the input block, the input x is mapped to a degree of freedom of a physical system. The experimental block includes the actual preparation of the corresponding initial state, a nonlinear time evolution, and the readout. Finally, in the output block, the physical readout is binned and a trained linear transformation produces the output. (b) An input value x_i is mapped to the phase function $\phi(z)$ of an elongated Bose-Einstein condensate. The phase imprint is realized using a digital micromirror device (DMD). After the time evolution, the density distribution is read out. The density is further processed by binning and transformed via a vector multiplication to a single output value.

the blue-detuned light on a specific area of the atomic cloud for 100 µs, inducing an estimated phase shift of 1.4π on one side of the cloud [see Fig. 1(b)]. Using the DMD, we imprint the phase step at arbitrary positions within the central area ($Z = 40$ µm) of the atomic cloud with excellent reproducibility. Careful adjustment of the light intensity for the box trap avoids changes of the trapping frequency along the short axis during the imprint. After one input value is encoded, the atomic cloud evolves freely in the box trap. The imprint leads to the formation of a density peak, which moves along the cloud and disperses, as well as a stationary but decaying density dip [see Fig. 3(a)]. The density profile is extracted after a variable evolution time t_{evo} via two-frequency absorption imaging [11]. The resulting image of the elongated density distribution is integrated along the short axis of the cloud.

IV. TRAINING THE LINEAR OUTPUT BLOCK

In the first layer of the linear output block, the values of the extracted density are binned, leading to a vector $\bar{\mathbf{n}}_{\mathbf{i}}$ of length N_w. Then, the final linear transformation is defined by supervised learning, minimizing the root-mean-square (RMS) error,

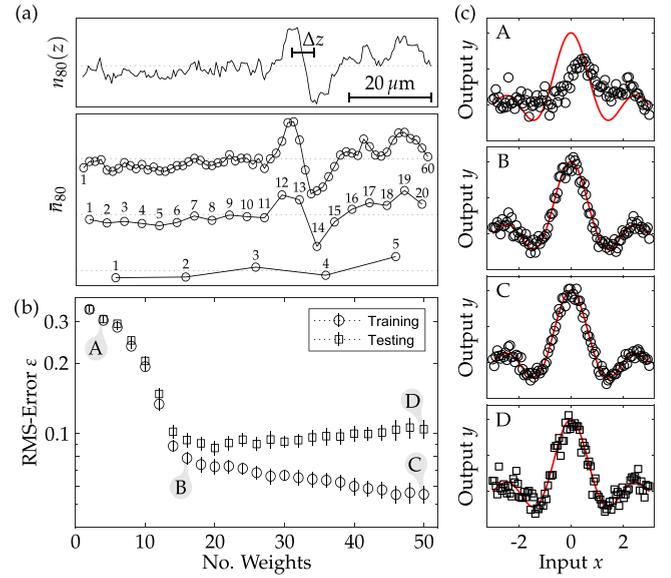$$\varepsilon = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - u_i)^2} \,, \qquad (1)$$

FIG. 2. Performance dependence on number of weights. (a) Density distribution for an evolution time of 1.1 ms after the phase imprint (top). Lowering the number of bins, N_w, reduces the spatial resolution and leads to less information about the original distribution, while reducing the noise (bottom). (b) RMS error of training and testing as a function of N_w. For training, the RMS error decreases for higher bin numbers, with a characteristic kink at $\sim$18 bins. This behavior is caused by the dependence of both expressivity and signal to noise on the number of bins. Error bars have been calculated by bootstrap resampling analysis (see the Appendix). (c) Examples for training (A–C) and testing (D) for different number of bins. The red lines indicate the target functions.

where $y_i \in \mathbb{R}$ are the results of the model and $u_i = f(x_i) \in \mathbb{R}$ are the prescribed function values to be learned. Since $N_w < N_x$, this is achieved by calculating the Moore-Penrose pseudo-inverse [12] for the system of linear equations

$$\begin{pmatrix} \bar{n}_{1,1} & \cdots & \bar{n}_{1,N_w} \\ \vdots & & \vdots \\ \bar{n}_{N_x,1} & \cdots & \bar{n}_{N_x,N_w} \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_{N_w} \end{pmatrix} = \begin{pmatrix} u_1 \\ \vdots \\ u_{N_x} \end{pmatrix} \,, \qquad (2)$$

constructed from measured $\bar{\mathbf{n}}_{\mathbf{i}}$ and target values u_i associated with the input values x_i. Note that the number of bins is also the number of weights, N_w. Therefore, there is an optimal N_w much smaller than N_x where high signal to noise as well as expressive power are given.

V. REGRESSION TASK

We solve a regression task for the function $f(x) = \sin(\pi x)/(\pi x)$, a standard nonlinear function for benchmarking. For that purpose, we compile the dataset $\{(x_i, u_i)\}$ from $N_x = 100$ evenly spaced input values $x_i \in [-3, 3]$ with their associated target values $u_i = f(x_i)$. The experiment is run for each input value x_i with the associated phase imprint and an evolution time of $t_{\text{evo}} = 1.1$ ms. An example of the resulting density distribution for the specific input value x_{80} is shown in the upper trace of Fig. 2(a). The linear transformation is then trained by inverting Eq. (2) with binned density profiles $\bar{\mathbf{n}}_{\mathbf{i}}$ [see Fig. 2(a), lower panel], resulting in an optimized weight
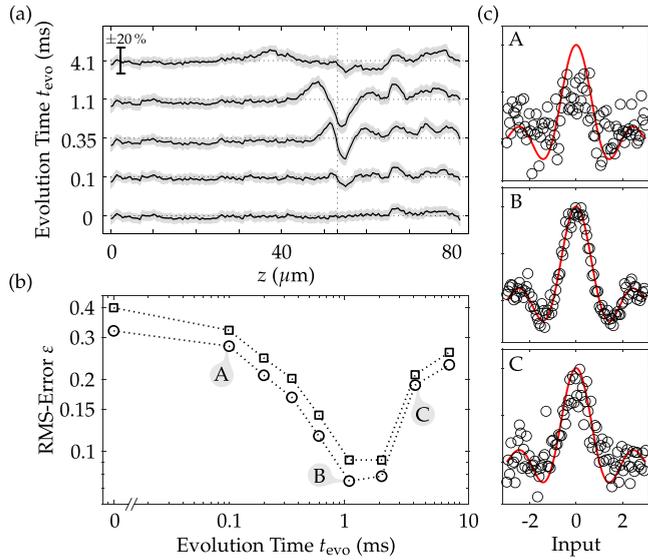
FIG. 3. Performance dependence on evolution time. (a) Density distributions after different evolution times. Depicted is the mean over 30 realizations, with the standard deviation given by the shaded area. Over time, the imprint leads to a density peak, which moves to the left and widens, and a nonmoving density dip. For evolution times longer than 2 ms, both features slowly decay. (b) RMS error of training and testing as a function of evolution time. For each evolution time, the number of weights, N_w, is chosen such that the RMS error of testing is minimal. An optimum for both training and testing can be found at 1.1 ms, where the signal to noise of the binned density profiles is maximal. (c) Examples for training (A-C) for evolution times. The red lines indicate the target functions.

vector **w**. This training is successful if the weight vector leads to viable results not only for training data, but also for new experimental realizations of the same input values (testing).

Training on differently binned data reveals that there is an optimal binning giving the best performance, as can be seen in Fig. 2(b). The RMS error for training monotonically decreases with the number of weights, N_w, and ultimately vanishes as the matrix in Eq. (2) reaches full rank, i.e., the number of inputs equals the numbers of weights. A characteristic kink at ~18 weights can be identified. While the RMS error of the training suggests better performance with a further increase of N_w, testing clearly reveals that the performance of the regression does not improve beyond the kink, but saturates and ultimately decreases. As can be seen in Fig. 2(a), the number of bins, N_w, affects the spatial resolution. Thus the system's expressivity is limited not only by the number of weights, N_w, but also by the information extracted from the physical system. A minimal sampling is required to resolve the density feature, and therefore the substantial output information is accessed given $N_w > 2Z/\Delta z = \tilde{N}_w$, where Δz is the characteristic extension of the density feature and Z is the length of the region used for information encoding, in close analogy to the Nyquist-Shannon sampling criterion [13].

The deviation of the performance of training and testing occurs because the training becomes dominated by a generic property of a physical system—the noise. For ultracold gases, this is ultimately given by the shot noise due to the finite number of particles. With binning, this noise level is reduced

since adjacent pixels are averaged. Because fewer pixels are averaged as the number of bins increases, the noise on the density profiles increases as well. This increase of noise can be partially compensated in the training data due to the increasing number of weights, i.e., expressivity; however, when applying the trained weights to testing data, the noise directly translates into poorer performance. As a consequence, the best performance of the experimental block strongly depends on N_w, which has to be optimized to balance information extraction and noise reduction. This is system specific and has to be determined for a chosen physical system. In machine learning, this phenomenon is known as the bias-variance trade-off or decomposition [10, Chap. 3.2].

VI. TIME EVOLUTION OF THE PHYSICAL SYSTEM

To analyze the relevance of the physical evolution, the quality of the regression is investigated for different evolution times. Figure 3(a) shows the averaged (30 realizations) density profiles corresponding to the input value x_{80} for different evolution times. The profiles share the same scale and are shifted by a constant offset for clarity. The standard deviation of the mean is given by the gray shaded area. After the preparation, a peak of expelled atoms moves to the left while the depletion remains stationary. Similar schemes of phase imprinting have been employed for dark soliton preparation [14,15].

For a quantitative analysis, the model is trained with different evolution times of the physical system. The resulting RMS error ε with N_w chosen optimally at each evolution time is shown in Fig. 3(b) for training and testing. With increasing evolution time, the error decreases and reaches a minimum at $t_{evo} = 1.1$ ms. This behavior can be understood from the evolution of the profiles, where the density features resulting from the phase imprint have to develop first and widen. For times shorter than 1.1 ms, the narrow density features imply that the optimal number of bins, $\tilde{N}_w \propto Z/\Delta z$, is high. Thus the resulting signal-to-noise ratio is low, limiting the performance. The increase in RMS error for significantly longer times is mainly a result of the decreasing signal due to the spreading of the peak and filling of the depletion. The testing error is close to the training error for all times, demonstrating successful learning (in contrast to overfitting; see the Appendix). The significant reduction of the error for evolution times of the order of 1 ms indicates a high predictive power of our model.

VII. GENERALIZATION

To estimate the model's ability of generalization, we test the performance with input values x_i that are not part of the training data set. We define good performance of a regression task as the robust and reliable interpolation between the given supporting points x_i. For the following discussion, we choose the derivative of the former target function $f(x) = \sin(\pi x)/(\pi x)$ since it is better suited to reveal the direct connection between our machine learning model and linear basis function models [10, Chap. 3.1]. We use the experimental block at its best performance given by $t_{evo} = 1.1$ ms and binning of $N_w = 18$, and take the $N_x = 100$ inputs equally spaced in the range of $x_i \in [-1.5, 1.5]$. We investigate the performance of the interpolation both for equally spaced
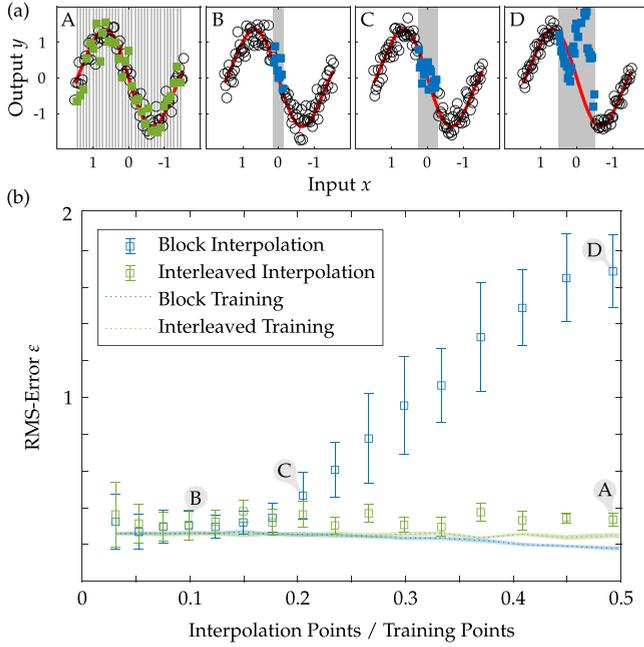
FIG. 4. Generalization. The model is used for interpolation between trained input values. (a) Examples of the performance. Red line: Target function; circles: training points; squares: interpolation points. The interpolation inputs are either equally spaced over the whole interval (interleaved case) or all from the central region (block case). The interpolation regions are indicated by gray shading. The interpolation between adjacent training points (interleaved) works very well (A), but fails beyond a critical block length (B)–(D). (b) Dependency of the RMS error on the ratio between interpolation and training points. The interleaved case works well for all ratios. For ratios beyond 0.2, corresponding to a block size of 17 interpolation points, the performance drastically decreases, which can be understood as the failure of the corresponding wavelet decomposition (for details, see text). Error bars are the result of bootstrap resampling analysis.

interpolation points over the whole input range (interleaved) and for interpolation points in a single domain (block) [see Fig. 4(a)]. Panel (A) shows the good performance in interpolating every third input qualitatively. This is confirmed by the RMS error [see Fig. 4(b)], which hardly depends on the distance between interpolated points. Block interpolation, however, works only up to a certain block size [panels (B)–(D)]. The RMS error grows beyond a ratio between interpolation points and training points of 0.2, corresponding to block sizes larger than $\Delta x = 17$ input points. This behavior can be understood by reinterpreting the data matrix in Eq. (2) to

$$
\begin{pmatrix} - & \bar{n}_1 & - \\ & \vdots & \\ - & \bar{n}_{N_x} & - \end{pmatrix} \equiv \begin{pmatrix} | & & | \\ g_1 & \cdots & g_{N_w} \\ | & & | \end{pmatrix}, \qquad (3)
$$

where the introduced $(g_j)_i$ is the mean density in bin j given input x_i. From this perspective, the linear transformation in the output block composes the discretized function $f(x_i) = u_i$ from the limited number of $(g_j)_i$ [16, Chap. 4]. This is the discretized version of a decomposition of the target function

into a finite set of representative (basis) functions,

$$
f(x) = \sum_{j=1}^{N_w} w_j \, g_j(x), \qquad (4)
$$

which is known as wavelet analysis and is at the heart of linear basis function models. Our physical system produces basis functions of spatially localized dispersive signals $g_j(x)$. Due to the diagonal nature of our data matrix (see the Appendix), the relative width of the basis function $\Delta x/X$, where X is the range of x values, is similar to the relative width of the density features in $\bar{\mathbf{n}}_\mathbf{i}$, $\Delta z/Z$. Interpolation does not work if the basis function corresponding to the center of the interpolation region has vanishing overlap with the basis functions corresponding to the edges of the block. This explains the failure of interpolation beyond extended regions of $N_x \Delta z/Z \sim 17$ interpolation points, which is equivalent to a ratio of interpolation to training points of 0.2 [see Fig. 4(b)].

VIII. CONCLUSION AND OUTLOOK

The demonstration of a machine learning model with a highly controlled Bose-Einstein condensate as the nonlinear experimental block opens the general perspective of fusing highly controlled physical systems with classical computing. The approach and results are agnostic to the underlying theoretical description of the ongoing physics, but rely on precise experimental control. We have chosen the regression problem for the first demonstration since the performance can be straightforwardly quantified. It also clearly demonstrates the parallels between our machine learning model and the paradigmatic linear basis function model. Since our physical system offers a broad range of precisely adjustable parameters (plasticity), such as nonlinearity (interaction strength), potential landscapes, and temperature, the input block as well as the experimental block could be optimized for specific tasks. In particular, the perspective of utilizing quantum features of the physical system in machine learning architectures opens up different possibilities for information processing.

APPENDIX

1. Experimental system

We prepare a Bose-Einstein condensate of ^{39}K with approximately 20 000 atoms in the substate corresponding to $F = 1$, $m_F = -1$ at low magnetic fields. The scattering length is tuned to $50\,a_B$ (a_B is the Bohr radius) by applying a homogeneous magnetic field to exploit the Feshbach resonance at 561 G [17,18]. An additional magnetic gradient levitates
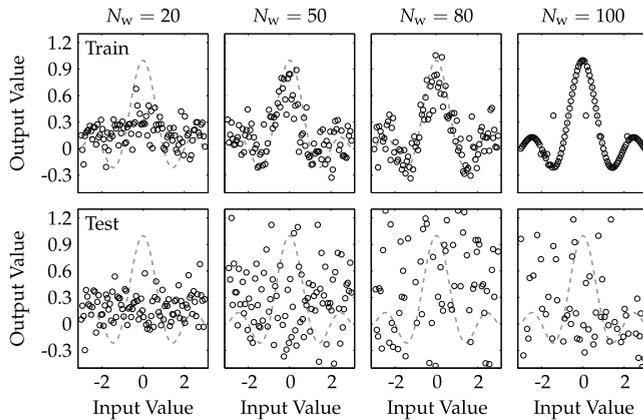
FIG. 5. Example of overfitting. The top four plots show the result of training a regression, where reference profiles were used for each input value ($N = 100$). They are completely uncorrelated to the input as they are obtained without the phase imprint. However, by increasing the number of weights, N_w, a regression to the target function (dashed line) seems possible. Testing the obtained weight vector w with a different set of inputs reveals that no prediction can be made and the results stem from overfitting (lower row).

the atoms against gravity. In the vertical direction, the atomic cloud is confined in a single lattice site of a repulsive lattice created by a blue-detuned laser beam (532 nm light, lattice spacing 5 μm). This leads to a strong two-dimensional confinement with a trap frequency of $\omega \sim 2\pi \times 1.5$ kHz. In the horizontal plane, a configurable dipole potential is applied with a blue-detuned 532 nm laser beam, which is shaped by a digital micromirror device (DMD) in direct imaging configuration [19]. To avoid uncontrolled interferences between vertical and horizontal confinement, the light frequencies are shifted by 160 MHz. The DMD also allows one to imprint phase shifts on the atoms by illuminating one area of the cloud. The density distribution of the atomic cloud is extracted by absorption imaging with a two-frequency scheme [11]. The resolution for both setups is ~ 1 μm. We estimate the chemical potential from the velocity of the density peaks to be ~ 1.2 kHz and the healing length to be ~ 0.3 μm. From images of the elongated cloud after an evolution of a quarter period in a slow harmonic trap (≈ 5 Hz) [20], we confirm that the temperature of the cloud is well below the critical temperature T_c.

2. Bootstrap resampling analysis

In order to estimate the statistical deviation of the RMS error with a limited number of experimental runs, we employ a method inspired by bootstrap resampling [21]. For every input value x_i (where $i = 1, \ldots, N$), we measure N_R different density profiles $n_{i,k}$. We then compile a set of density profiles, $\mathcal{N}_{\text{Train}}$, by randomly choosing one of the realizations,
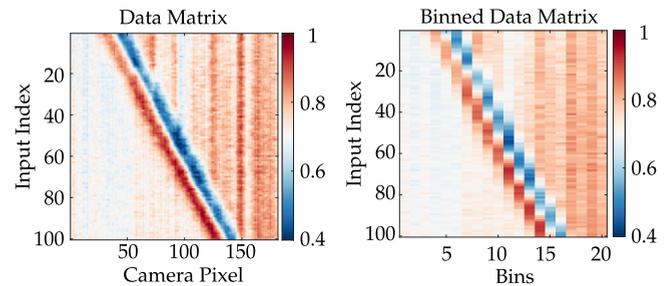

FIG. 6. Data matrix. Left: Bare data. Right: Binned data.

$k_i = 1, \ldots, N_R$, for every input value x_i,

$$\mathcal{N}_{\text{Train}} = \{n_{1,k_1} \ldots n_{N,k_N}\}. \tag{A1}$$

For this set of experimental realizations, the training is applied to obtain the weight vector **w**. We then calculate the RMS error ε for the training set according to Eq. (1) to judge the quality of the regression. Next, a testing set $\mathcal{N}_{\text{Test}}$ is put together in the same way as the training set $\mathcal{N}_{\text{Train}}$, but using only density profiles $n_{i,k} \notin \mathcal{N}_{\text{Train}}$. Then, the RMS error ε is calculated for the testing set $\mathcal{N}_{\text{Test}}$. This technique allows the generation of many different training sets from a limited number of experimental realizations per input value. The results for the RMS error shown in this publication are calculated from the mean and standard deviation of 25 repetitions of this process.

3. Overfitting

To illustrate the importance of testing the obtained weight vector **w**, we sabotage the training procedure by using reference profiles as the input profiles n_i. These noisy but flat profiles do not have any phase imprinted and are thus uncorrelated to input values x_i. However, by increasing the number of weights, N_w, a fit becomes possible (Fig. 5, upper row). If the number of parameters matches the number of training points and all input profiles are linearly independent, then Eq. (2) becomes invertible and the training data are matched. Since there is no information encoded in the data, testing with data previously unseen by the model will fail. In the lower row of Fig. 5, this lack of predictive power is obvious.

4. Data matrix

The one-dimensional data vectors for all input values and $t_{\text{evo}} = 1.1$ ms are shown in Fig. 6, in the left plot. Each row is the detected one-dimensional density n_i on the camera for input value x_i. One sees the emergent density peak and dip. The position of the pattern is linearly dependent on the input value, and thus $\Delta x/X \propto \Delta z/Z$. On the right side of the phase step, small artifacts in the density are produced when the DMD is switched on to imprint a phase on one portion of the cloud. The right plot shows the data set binned for optimal performance, i.e., 20 bins.

[1] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks – with an erratum note, German Nat. Res. Cent. Inf. Technol. GMD Tech. Rep. **148**, 13 (2001).

[2] T. Natschläger, W. Maass, and H. Markram, The "liquid computer": A novel strategy for real-time computing on time series, Telematik **8**, 39 (2002).

[3] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, An experimental unification of reservoir computing methods, Neural Networks **20**, 391 (2007).

[4] M. Lukoševičius and H. Jaeger, Reservoir computing approaches to recurrent neural network training, Comput. Sci. Rev. **3**, 127 (2009).

[5] K. Nakajima, Physical reservoir computing—An introductory perspective, Jpn. J. Appl. Phys. **59**, 060501 (2020).

[6] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, Recent advances in physical reservoir computing: A review, Neural Networks **115**, 100 (2019).

[7] K. Nakajima and I. Fischer, *Reservoir Computing: Theory, Physical Implementations, and Applications* (Springer, Singapore, 2021).

[8] G. Marcucci, D. Pierangeli, and C. Conti, Theory of neuromorphic computing by waves: Machine learning by rogue waves, dispersive shocks, and solitons, Phys. Rev. Lett. **125**, 093901 (2020).

[9] N. A. Silva, T. D. Ferreira, and A. Guerreiro, Reservoir computing with solitons, New J. Phys. **23**, 023013 (2021).

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed., Information Science and Statistics (Springer, New York, 2006).

[11] M. Hans, F. Schmutte, C. Viermann, N. Liebster, M. Sparn, M. K. Oberthaler, and H. Strobel, High signal to noise absorption imaging of alkali atoms at moderate magnetic fields, Rev. Sci. Instrum. **92**, 023203 (2021).

[12] R. Penrose, On best approximate solutions of linear matrix equations, Math. Proc. Cambridge Philos. Soc. **52**, 17 (1956).

[13] C. Shannon, Communication in the presence of noise, Proc. IRE **37**, 10 (1949).

[14] S. Burger, K. Bongs, S. Dettmer, W. Ertmer, K. Sengstock, A. Sanpera, G. V. Shlyapnikov, and M. Lewenstein, Dark solitons in Bose-Einstein condensates, Phys. Rev. Lett. **83**, 5198 (1999).

[15] J. Denschlag, J. E. Simsarian, D. L. Feder, C. W. Clark, L. A. Collins, J. Cubizolles, L. Deng, E. W. Hagley, K. Helmerson, W. P. Reinhardt, S. L. Rolston, B. I. Schneider, and W. D. Phillips, Generating solitons by phase engineering of a Bose-Einstein condensate, Science **287**, 97 (2000).

[16] M. Hardt and B. Recht, *Patterns, Predictions, and Actions: Foundations of Machine Learning* (Princeton University Press, Princeton, NJ, 2022).

[17] J. Etrych, G. Martirosyan, A. Cao, J. A. P. Glidden, L. H. Dogra, J. M. Hutson, Z. Hadzibabic, and C. Eigen, Pinpointing Feshbach resonances and testing Efimov universalities in 39K, Phys. Rev. Res. **5**, 013174 (2023).

[18] C. D'Errico, M. Zaccanti, M. Fattori, G. Roati, M. Inguscio, G. Modugno, and A. Simoni, Feshbach resonances in ultracold 39K, New J. Phys. **9**, 223 (2007).

[19] G. Gauthier, I. Lenton, N. M. Parry, M. Baker, M. J. Davis, H. Rubinsztein-Dunlop, and T. W. Neely, Direct imaging of a digital-micromirror device for configurable microscopic optical potentials, Optica **3**, 1136 (2016).

[20] S. Tung, G. Lamporesi, D. Lobser, L. Xia, and E. A. Cornell, Observation of the presuperfluid regime in a two-dimensional Bose gas, Phys. Rev. Lett. **105**, 230408 (2010).

[21] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap* (Chapman and Hall, New York, 1994).