Estimating physical properties from liquid crystal textures via machine learning and complexity-entropy methods

H. Y. D. Sigaki,[1] R. F. de Souza,[1] R. T. de Souza,[1,2] R. S. Zola,[1,2,*] and H. V. Ribeiro[1,†]
[1]*Departamento de Física, Universidade Estadual de Maringá, Maringá, PR 87020-900, Brazil*
[2]*Departamento de Física, Universidade Tecnológica Federal do Paraná, Apucarana, PR 86812-460, Brazil*

(Received 12 September 2018; published 22 January 2019)

Imaging techniques are essential tools for inquiring a number of properties from different materials. Liquid crystals are often investigated via optical and image processing methods. In spite of that, considerably less attention has been paid to the problem of extracting physical properties of liquid crystals directly from textures images of these materials. Here we present an approach that combines two physics-inspired image quantifiers (permutation entropy and statistical complexity) with machine learning techniques for extracting physical properties of nematic and cholesteric liquid crystals directly from their textures images. We demonstrate the usefulness and accuracy of our approach in a series of applications involving simulated and experimental textures, in which physical properties of these materials (namely: average order parameter, sample temperature, and cholesteric pitch length) are predicted with significant precision. Finally, we believe our approach can be useful in more complex liquid crystal experiments as well as for probing physical properties of other materials that are investigated via imaging techniques.

DOI: 10.1103/PhysRevE.99.013311

I. INTRODUCTION

Optical imaging techniques are important tools extensively used for probing a number of materials properties [1]. These imaging techniques are nondestructive and particularly convenient for dealing with biological and other complex materials [2]. Liquid crystals are among these materials widely studied via optical and image processing methods [3], because liquid crystals are birefringent materials, and as such, simple polarized optical microscope imaging already access some of their important properties, including birefringence and sample thickness [4]. Moreover, this technique estimates the local ordering properties (for instance, the director distribution) across a sample when coupled with variable retarders and different algorithms for fast and sensitive measurements [5]. This approach is known as LC-PolScope [6] and has been used for fine imaging of defect cores in lyotropic liquid crystals [7] and can describe the orientational order of active nematics [8].

Despite the extensive use of optical imaging approaches in the study of liquid crystals [9–12], much less attention has been paid to the problem of extracting physical parameters directly from images of these materials. This is an important issue since several physical parameters of liquid crystals are obtained only by adjusting theoretical models to cumbersome and time-demanding experimental results. Examples include the microscopic order parameter, from which several other parameters characterizing the nematic phase are dependent [3], and the pitch length of cholesteric liquid crystals. The latter is easily obtained under an optical microscope when the helical axis lies perpendicular to the viewing direction [13] but cannot be estimated from the most commonly used experimental arrangements, where the helix orients parallel to the viewing direction (often called Grandjean texture, used in reflective displays) [14].

In this context, image-based characterization of liquid crystals can benefit hugely from state-of-the-art machine learning techniques [15]. These approaches have been available since the 1990s, but it was only during the last decade that such methods gained impressive popularity in several areas of science where unveiling meaningful patterns in data is fundamental. Naturally, physics is not an exception, and indeed there are several recent works employing machine learning algorithms for studying many physical systems [16–24]. Here we present an approach that is capable of extracting physical properties of nematic and cholesteric liquid crystals directly from their textures images. Our approach is based on the evaluation of two complexity measures related to the arrangement of pixels in the textures, combined with simple machine learning algorithms employed for classification and regression tasks. We demonstrate the potential of this approach in a series of applications based on numerically generated and experimental textures, from which physical parameters of liquid crystals are predicted with high accuracy.

In what follows, we present our results on simulated and experimental nematic liquid crystals, where we show that our approach predicts the average order parameter from simulated textures with an accuracy of ≈99% and the sample temperature with an accuracy of ≈93%. We further show that this approach classifies different pitches of cholesteric textures with an overall accuracy of ≈85%. Next, we present our conclusions and methodological details of the proposed approach as well as our experimental procedure and numerical methods used for obtaining liquid crystal textures.

[*] rzola@utfpr.edu.br
[†] hvr@dfi.uem.br

II. RESULTS

In all studies presented here, we have calculated two simple complexity measures directly obtained from liquid crystal textures: the permutation entropy H and the statistical complexity C [25–28]. As detailed in Appendix A, these two quantities are estimated from an ordinal probability distribution $P = \{p_1, p_2, \ldots, p_n\}$, whose components represent the probability of finding a given two-dimensional local ordering pattern of size $d_x \times d_y$ (the embedding dimensions) over the pixels of a texture. The entropy H quantifies the degree of disorder regarding the occurrence of these local patterns. A texture characterized by $H \approx 1$ has pixels randomly distributed in space, while for $H \approx 0$, these pixels appear practically at the same order along the texture. The complexity C quantifies the degree of "structuredness" in the arrangement of pixels in a texture. A value of $C \approx 0$ occurs for both extremes of order and disorder, whereas $C > 0$ represents a texture with more complex spatial patterns.

We employ only these two measures as predictive features of physical properties of liquid crystals. The predictions are obtained by training the nearest neighbors algorithm [15] with the values of H and C from a set of textures in which the physical property in question is known, and next the trained algorithm is exposed to values of H and C from another set of textures in order to predict the physical property. The nearest neighbors algorithm is one of the simplest machine learning algorithms for classification and regression tasks [15], but it already provides excellent accuracy in our results. Details of the implementation of this algorithm are given in Appendix E.

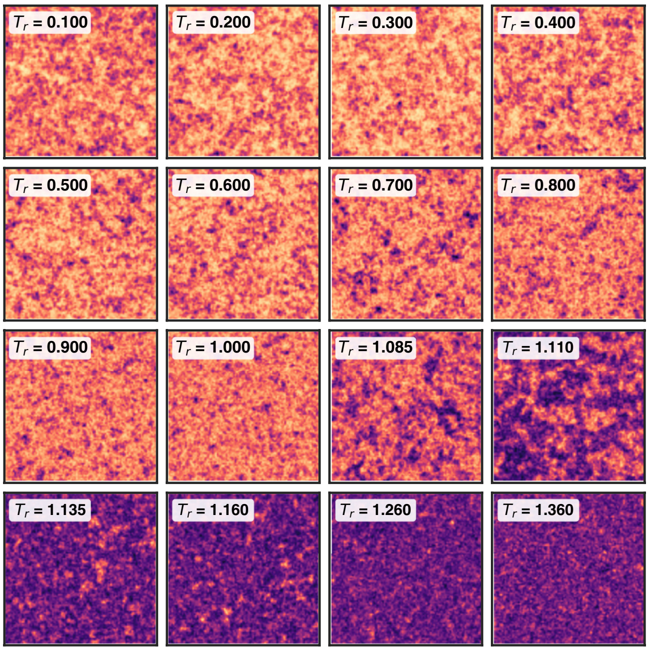

FIG. 1. Examples of a nematic liquid crystal texture at different temperatures and phases. These textures are generated by using the Monte Carlo method described in Appendix B. Each texture corresponds to a different reduced temperature T_r (as indicated within the plots), and this system presents a nematic to isotropic phase transition at the critical temperature $T_c = 1.1075$. We note that textures from different phases are easily distinguishable, while nematic textures ($T_r < T_c$) from different reduced temperatures are very similar to each other as well as the ones from the isotropic phase ($T_r > T_c$).

A. Monte Carlo simulated textures

We start by analyzing nematic textures generated by Monte Carlo simulations of the model described in Appendix B. Examples of these textures are shown in Fig. 1 for different reduced temperatures T_r. This system undergoes a nematic to isotropic phase transition when T_r exceeds the critical temperature $T_c = 1.1075$. Each texture has a different average order parameter p that depends on the temperature T_r [see Eq. (B3) for details], and our goal is to predict the value of p directly from these images by using the values of H and C. We note that the textures exhibit visually distinct patterns between the nematic and isotropic phases; in particular, for $T_r > T_c$ we observe the emergence of isotropic domains that predominate in the texture as the temperature increases beyond T_c. While nematic and isotropic textures are easily distinguished from each other, even a well-trained eye of an experimental physicist will be in trouble for distinguishing among nematic textures with different temperatures (for instance, between $T_r = 0.2$ and $T_r = 0.6$) as well as among isotropic textures at different temperatures.

Despite the visual similarity among the textures, we show that the values of H and C are capable of distinguishing among these images as well as identifying the nematic-isotropic transition. To do so, we create a data set composed of several realizations of simulated nematic textures for different temperatures T_r, and evaluate the values of H and C for each one. Figures 2(a) and 2(b) show the dependence of the average values of H and C on the reduced temperature T_r. We observe that H has a trend to increase with the temperature but shows a sharp minimum at $T_r = T_c$. Similarly, the values of C tend to decrease with the temperature and present a sharp maximum at $T_r = T_c$. Figure 2(c) depicts the dependence of the order parameter p on the reduced temperature T_r, where the critical temperature $T_c = 1.1075$ is defined as the value that maximizes the absolute value of the derivative of p with respect to T_r.

The well-defined dependence of the average values of H and C on the temperature T_r, combined with the fact that the order parameter p is also a function of T_r, indicates that we can predict the values of p directly from the images. It is worth noting that the values of H and C are not uniquely defined for a given temperature, displaying some random fluctuations associated with the process that generates these textures. Thus, to test the predictive power of these image quantifiers in a more practical situation, we have trained a k-nearest neighbors algorithm for the regression task of predicting the order parameter p based on the values of H and C and a dummy variable that is zero for $T_c - 0.05T_c \leqslant T_r \leqslant T_c$, 1 for $T_r > T_c$, and -1 for $T_r < T_c - 0.05T_c$. This dummy variable is necessary due to the nonbiunivocal relations between the image quantifiers and the temperature. However, the value of T_c can be directly estimated from the image quantifiers. As detailed in Appendix E, the k-nearest neighbors algorithm is among the most straightforward statistical learning approaches that assigns to an unlabeled object (the value of p

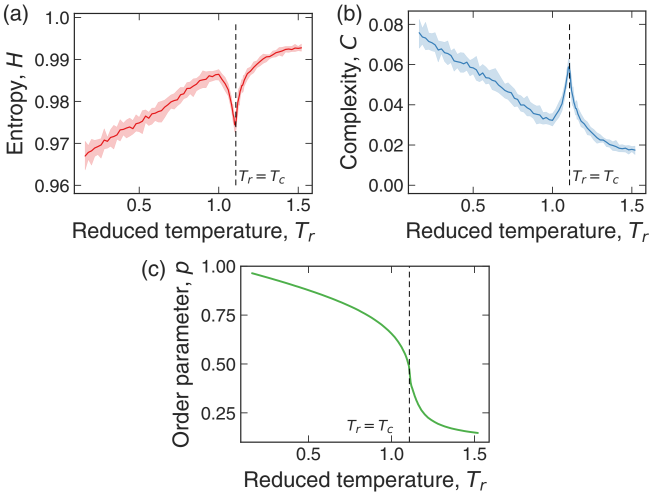

FIG. 2. Dependence of the image quantifiers and the order parameter on the temperature. (a) Values of the permutation entropy H and (b) the statistical complexity C as a function of the reduced temperature T_r. The solid curves represent average values over 50 realizations, and the shaded areas are the 95% bootstrap confidence intervals. (c) Dependence of the order parameter p versus the reduced temperature T_r. In all panels, the dashed line indicates the critical temperature ($T_c = 1.1075$) of the nematic-isotropic phase transition. We note that the phase transition is clearly and properly identified by the extreme values of the complexity measures.

from a texture) the most frequent label among the k training nearest neighbors (the parameter of the method) in the features space (the extracted image features).

Figure 3(a) shows the validation curves, that is, the training and cross-validation scores as a function of the number of neighbors k. We note that the algorithm overfits the data for $k < 10$, that is, the algorithm is not complex enough to capture the underlying structure of the data. On the other hand, the algorithm begins to underfit the data for $k > 20$, meaning that the learning method is getting too complex and modeling even the random noise in the training set. Figure 3(b) shows the learning curves, which represents the scores as a function of the fraction of the data used for training the algorithm (with $k = 15$). We observe no significant improvement in the cross-validation score when more than 35% of the data is used as training set. Thus, the k-nearest neighbors algorithm achieves a remarkable accuracy of ≈99.2% in the regression task of predicting the order parameter p solely based on the values of H and C. We further observe that practically the same accuracy is obtained when using only the values of H or only the values of C. This happens because the values of H and C are strongly correlated to each other for these textures. However, in general, the values of C are not a trivial function of H and usually care additional information related to the "structural" complexity of images [27–29]. The performance of this algorithm is much higher than those obtained from simple baseline regressors that always predict the expected values or the median (accuracy of ≈0%). Figure 3(c) shows a comparison between the actual values of the order parameter p as a function of the reduced temperature T_r in a particular simulation and the values predicted by the k-nearest neighbors algorithm (with $k = 15$). We note that the predictions are very close to the actual values of the order parameter, which confirms the efficiency and usefulness of our approach.

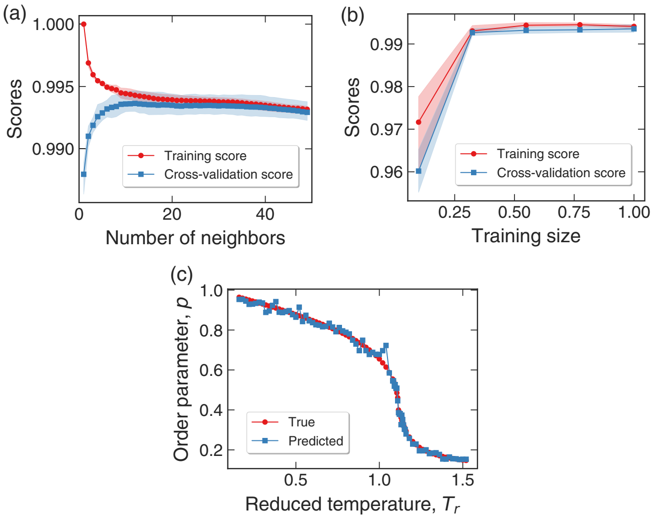

FIG. 3. Predicting the order parameter p with a machine learning algorithm. (a) Training and cross-validation scores of the k-nearest neighbors algorithm as a function of the number of neighbors k. The scores represent the coefficient of determination (R^2) of the relationship between the predicted and true values and can be interpreted as the percentage (or fraction) of data variation that is explained by the model. We note that the algorithm overfits the data when $k < 10$, whereas for $k > 20$ it starts underfitting the data. (b) Learning curves, that is, training and cross-validation scores as a function of the training size (fraction of the whole data used for training the model) with $k = 15$. We observe no significant improvement in the cross-validation score when more than 35% of the data is used. The shaded areas in both plots represent the 95% confidence intervals obtained in a three-fold-cross-validation splitting strategy. We note the high accuracy achieved by the algorithm (≈99.2%). (c) True and predicted order parameter p as a function of the reduced temperature T_r. These predictions were generated by exposing the trained regressor (with $k = 15$) to a set of textures never presented before to the algorithm.

B. Nematic textures of experimental samples

In another application, we study nematic textures obtained from samples of the E7 liquid crystal, a multicomponent mixture composed by cyanobiphenyl and cyanoterphenol that is commonly employed in the industry for producing displays. This liquid crystal exhibits a nematic-isotropic transition at $T_c \approx 58\,^{\circ}$C [4]. Details about the experimental procedures are provided in Appendix C, but it basically consists in using polarized optical microscope imaging for taking pictures of these textures at different temperatures T. Figure 4(a) shows examples of textures obtained from a sample with temperature varying from 40 °C to 60 °C. We observe that there are no visually significant changes in the patterns of these textures when the temperature increases until 55 °C. As the sample temperature exceeds the critical temperature T_c, we observe the growth of isotropic domains which makes easier the visual distinction among these textures.

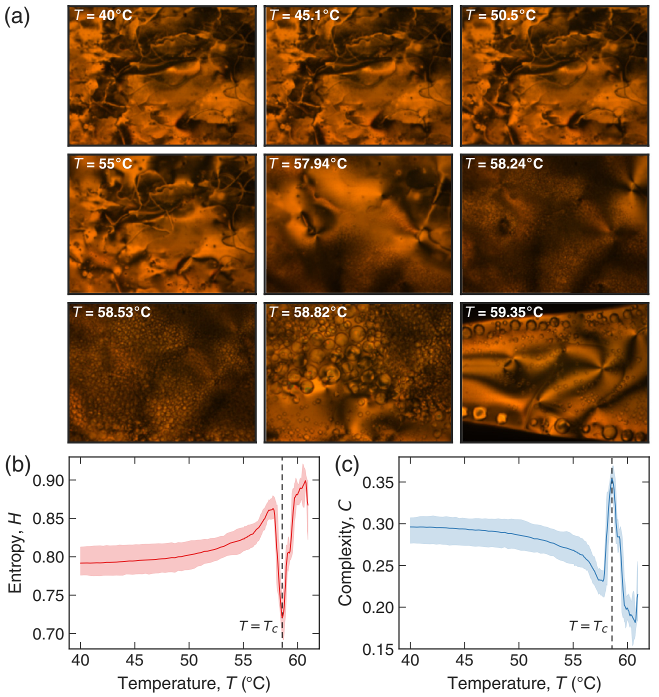

FIG. 4. Examples of E7 liquid crystal textures at different temperatures and phases, and the dependence of the complexity measures on the temperature. (a) Experimental textures of an E7 liquid crystal sample at different temperatures. We observe practically no visual change until the critical temperature $T_c \approx 58\,^{\circ}$C. (b) Dependence of the permutation entropy H and (c) the statistical complexity C on the temperature T. The solid curves represent the average of the quantities calculated over the results of six samples. The shaded areas are 95% bootstrap confidence intervals. The vertical dashed lines indicate the critical temperature T_c. We note that the phase transition is properly identified by the extreme values of the complexity measures.

We collect data from six different E7 samples by following the same experimental protocol. Figures 4(b) and 4(c) depict the average behavior of the entropy H and complexity C on the temperature T. These results are similar to those obtained via Monte Carlo simulation [see Fig. 2(a)]; that is, H tends to increase with T and shows a sharp minimum at the critical temperature, while C has a decreasing trend with T and displays a maximum at the critical temperature. Once again, we observe that the values of H and C are well-defined functions of the temperature T and capable of precisely identifying the nematic-isotropic transition.

Differently from our previous results on the simulated textures, we now propose to predict the sample temperature T (instead of the order parameter p) directly from the experimental textures. However, it is worth mentioning that in this simple experimental setup the order parameter can be estimated from the temperature [30], so that predicting the temperature is comparable to predicting the order parameter. We also note that most of the models for liquid crystals fail to predict the order parameter for temperatures much lower than T_C [30], and that in general, measuring the order parameter requires complicated procedures, such as light scattering experiments [31]. In order to perform these predictions, we proceed as in the simulated case, that is, we train a k-nearest neighbors algorithm for the regression task of predicting the values of T from the image quantifiers (H and C) and from a dummy variable that is zero for $T_c - 0.05T_c \leqslant T \leqslant T_c$, 1 for $T > T_c$, and -1 for $T < T_c - 0.05T_c$. Figure 5(a) shows the validation curves, where we observe that the algorithm overfits the data for $k = 1$, while it starts to underfit the data for k greater than 3. Figure 5(b) depicts the learning curves for $k = 2$, where we note no significant improvement in the cross-validation score when the training set exceeds 80% of the whole data. Thus, the k-nearest neighbors algorithm achieve an accuracy of ≈93% with $k = 2$ and by using 80% of data as training set. These scores are reduced in ≈6% when using the values of H and C separately, which reinforce the fact these complexity measures are not related to each other in a trivial manner. Furthermore, these scores are much higher than those obtained from regressors that always predict the expected values or the median (accuracy of ≈0%). Figure 5(c) illustrates the accuracy of these predictions by showing a scatter plot between the predicted temperature values and the actual values. We note that this relationship is

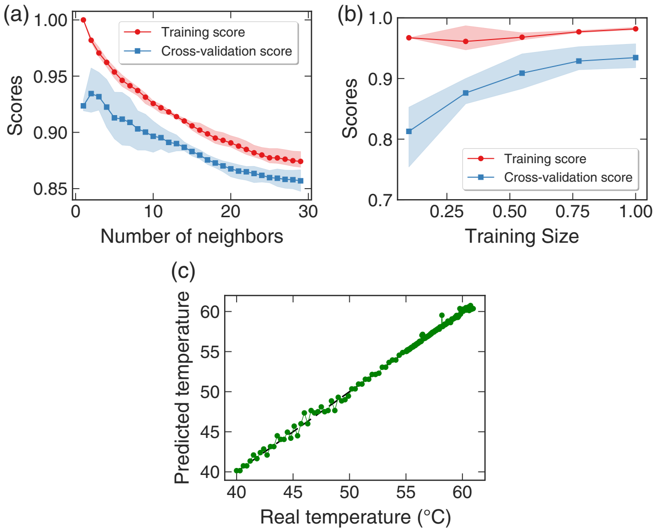

FIG. 5. Predicting the temperature with a machine learning algorithm. (a) Training and cross-validation scores of the k-nearest neighbors algorithm as a function of the number of neighbors k. The scores represent the coefficient of determination (R^2) of the relationship between the predicted and true values and can be interpreted as the percentage (or fraction) of data variation that is explained by the model. Note that for $k = 1$ the algorithm overfits the data, while for $k > 3$ the model starts to underfit the data. (b) Learning curves, that is, the training and cross-validation scores as a function of the training size (fraction of the whole data) for $k = 2$. The shaded areas in both plots represent the 95% confidence intervals obtained in a three-fold-cross-validation splitting strategy. We observe practically no significant improvement in the cross-validation score when more than 80% of the data is used to train the model. (c) True versus predicted temperatures obtained by exposing the trained regressor to a set of experimental textures never presented before to the algorithm. The dashed line represents the 1:1 relationship. We observe an excellent agreement between the true and predicted temperatures, reinforcing the great accuracy achieved by the method (≈93%).

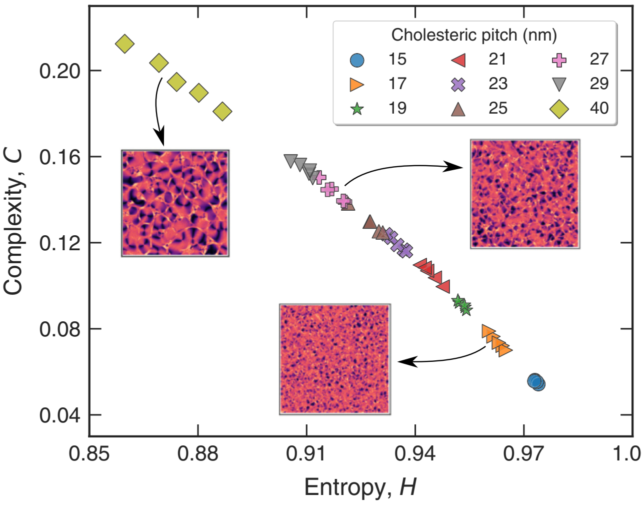

FIG. 6. Discriminating among cholesteric textures with different pitches via the complexity-entropy plane. Each colored marker represents the values of H and C for five realizations of the cholesteric textures with different pitches (as indicated by the different markers). We note that the values of H and C for each cholesteric pitch are localized over a small region in the complexity-entropy plane. We further observe that the textures from small pitches are localized in a high-entropy region, while those from large pitches are in a low-entropy region. This result indicates that textures from large pitches are more ordered than those obtained for small pitches (as illustrated by the insets).

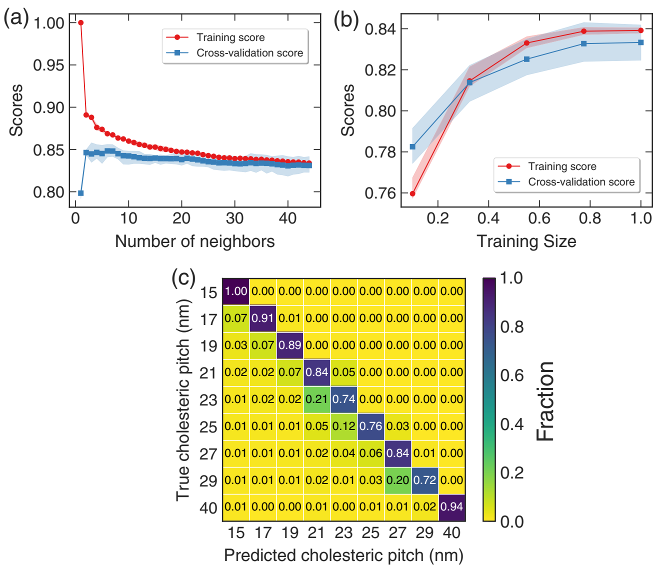

FIG. 7. Predicting the cholesteric pitch with statistical learning algorithms. Training and cross-validation scores (fraction of correct classifications) of the k-nearest neighbors algorithm as a function of the number of neighbors in (a) and as a function of the training size in (b). The shaded areas in both plots are the 95% confidence intervals obtained in a five-fold-cross-validation splitting strategy with the number of neighbors equal to 20. Note that for the number of neighbors smaller than 15 the algorithm overfits the data, but for more than 20 neighbors it starts underfitting the data. We also notice that when more than 60% of the data is used to train the model (training size), there is no significant score improvement. (c) True and predicted cholesteric pitch. This confusion matrix shows the good performance achieved by the algorithm represented by the high fractions of right predictions in the diagonal. In some cases, the algorithm underestimates the pitch. This results from the overlap observed in the complexity-entropy plane in Fig. 6.

an almost perfect 1:1 relation (indicated by the dashed line), which reinforces the overall quality of the k-nearest neighbors predictions and demonstrates the potential of our approach with experimental data.

C. Simulated cholesteric textures

As a last application, we investigate simulated textures of cholesteric liquid crystals. These materials display a helical structure composed of layers in between which the preferential director axis varies periodically with a period (that is, the distance to complete a full rotation of the director axis) known as the pitch η. Among other properties, the pitch of a cholesteric liquid crystal defines the wavelength of the reflected light as a consequence of the Bragg reflection in short pitch materials [3]. The pitch length modifies the textures of these materials, so that, a cholesteric texture can mimic a nematic one for large values of the pitch or be entirely different for short pitches. In a cell treated to impose homeotropic alignment, for example, the pitch length determines if the texture observed is homeotropic, fingerprint or focal-conic [30].

Our goal in this case is to identify the pitch of a cholesteric liquid crystal based on the values of H and C obtained from the textures. To do so, we create a dataset of textures composed of 100 replicas for each pitch value $\eta \in (15, 17, 19, 21, 23, 25, 27, 29, 40)$ nm, where $\eta = 40$ nm is large enough to mimic a nematic texture. The optical textures are numerically obtained by solving the model described in Appendix D, which is based on the Landau-de Gennes theory [32]. In our simulations, we have used real values for the physical parameters of this model but a small lattice size (see Appendix D for further details). This choice leads to unrealistic cholesteric pitches but generates textures very similar to those obtained from experimental results. In particular, we use the process of quenching a cholesteric sample from the isotropic state to generate the set of textures. Figure 6 shows the values of H and C estimated from five random selected textures for each value of the pitch as well as three typical textures for $\eta = 17$ nm, $\eta = 27$ nm and $\eta = 40$ nm (insets of that figure). We observe that although there exists some overlapping, textures with different pitches tend to occupy different regions on the complexity-entropy plane, indicating that H and C are capable of distinguishing among different cholesteric textures. We further note that the larger the values of η, the higher the complexity and the lower the entropy values. Thus, large values of pitch produce textures more locally ordered, while small values generate textures that are locally more irregular.

We train a k-nearest neighbors algorithm for the classification task of predicting the pitches solely based on the values of H and C. Figure 7(a) shows the validation curves. We observe that this algorithm overfits the data when the number of neighbors is smaller than 3, and for larger number of neighbors, it slowly starts underfitting the data. We highlight that this simple algorithm achieves an accuracy of $\approx 85\%$, which is much higher than the baseline accuracy ($1/9 \approx 11\%$) obtained from a classifier that makes uniformly random predictions. The scores of k-nearest neighbors algorithm are reduced in $\approx 5\%$ when using the values of H and C separately, similarly to what happens with the E7 nematic textures. Also, the learning curves depicted in Fig. 7(b) indicate that $\approx 60\%$ of the data is enough for fitting this algorithm to the cholesteric data. We have further estimated the confusion matrix, as shown in Fig. 7(c). The elements f_{ij} of this matrix represent the fraction of textures with pitch η_i that the algorithm predicts to have pitch η_j; thus, a perfect classifier is represented by an identity matrix ($f_{ij} = \delta_{ij}$). In practical applications, the closer to 1 are the diagonal elements of f_{ij}, the better is the performance of the classifier. In our case, we observe that nearly all nonzero elements of f_{ij} are within a diagonal band of width 1 of this matrix, with the main diagonal concentrating at least $\approx 72\%$ of the predictions. Thus, even when the algorithm incorrectly classifies the pitch of a texture (which occurs in about 15% of the predictions), it tends to predict a pitch value that is very close to the actual value, with a higher probability of underestimate the value (notice that the elements preceding the main diagonal are larger than those appearing after). These results thus corroborate to the usefulness of our approach for investigating more complex textures.

III. CONCLUSIONS

We have proposed an approach for extracting physical properties of liquid crystals directly from textures images of these materials. Our method is based on estimating two simple complexity measures (permutation entropy and statistical complexity) directly from the textures, which are used as features in supervised learning tasks of regression and classification of physical parameters of these materials. We have demonstrated the usefulness and accuracy of this approach in a series of numerical and experimental applications. Our results have shown that the average order parameter can be directly estimated from images of nematic textures obtained from Monte Carlo simulations with accuracy of 99%. Similar precision is obtained in the regression task of directly estimating the temperature from E7 liquid crystal textures at different temperatures and phases. We have further presented results based on cholesteric textures, in which we have probed the cholesteric pitch length with significant accuracy.

In spite of the significantly achieved accuracies, our approach is indeed quite simple and based on intuitive features very familiar to any physicist. Due to this underlying simplicity and also because our approach is very fast and scalable from the computational point of view, we believe it can be easily implemented and adapted for other more complex experimental situations involving the study of liquid crystals and perhaps for probing physical properties of different materials.

ACKNOWLEDGMENTS

This research was supported by CNPq and CAPES. HVR thanks the financial support of the CNPq under Grants No. 440650/2014-3, No. 303642/2014-9, and No. 407690/2018-2. R.S.Z. thanks the National Institute of Science and Technology Complex Fluids (INCT-FCx), and the São Paulo Research Foundation (FAPESP, 2014/50983-3).

APPENDIX A: COMPLEXITY-ENTROPY PLANE

The normalized permutation entropy H [25] and statistical complexity C [33] are two complexity measures originally proposed for characterizing time series [26], and that were more recently generalized for considering higher dimensional data such as images [27,28]. We refer the more detail-oriented reader to the previously cited references, where a complete description of these techniques can be found. Here we shall present both approaches through an illustrative example. For that, the matrix

$$A = \begin{bmatrix} 4 & 1 & 3 \\ 3 & 5 & 6 \\ 8 & 0 & 2 \end{bmatrix}$$

represents an hypothetical texture of size 3×3. The elements of this matrix indicate the light intensity transmitted through the sample around a particular site. For the experimental textures, these elements are obtained by averaging the shades of red, green, and blue of the image files in the RGB "color space" (see Appendix C for further details). We thus define sliding submatrices of size $d_x = 2$ by $d_y = 2$ (the embedding dimensions) in the form

$$A_i = \begin{bmatrix} a_0 & a_1 \\ a_2 & a_3 \end{bmatrix},$$

which for this particular example are

$$A_1 = \begin{bmatrix} 4 & 1 \\ 3 & 5 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 3 \\ 5 & 6 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 3 & 5 \\ 8 & 0 \end{bmatrix}, \quad \text{and} \quad A_4 = \begin{bmatrix} 5 & 6 \\ 0 & 2 \end{bmatrix}.$$

Next, we associate a sequence of symbols to each sub-matrix for representing the ordinal patterns of occurrence of their elements. In this case, A_1 is associated with $\Pi_1 = (1, 2, 0, 3)$ since $a_1 < a_2 < a_0 < a_3$, where Π_1 is the permutation that sorts the elements of A_1 in ascending order (line by line). Similarly, A_2 is represented by $\Pi_2 = (0, 1, 2, 3)$, since its elements are already in ascending order; A_3 is described by $\Pi_3 = (3, 0, 1, 2)$, since $a_3 < a_0 < a_1 < a_2$; finally, A_4 is associated with $\Pi_4 = (2, 3, 0, 1)$ because $a_2 < a_3 < a_0 < a_1$. In case of draws, the occurrence order of the tied elements is kept. By calculating the relative frequency of occurrence of each permutation, we estimate the probability distribution $P = \{p_i; i = 1, \dots, n\}$ for each one of the $n = (d_x d_y)!$ ordinal patterns. Among the $(d_x d_y)! = 24$ possible ordinal patterns, only four permutations have appeared once in our example, and therefore, $P = \{1/4, 1/4, 1/4, 1/4, 0, \dots, 0\}$ is the ordinal distribution associated with the matrix A.

The ordinal distribution $P = \{p_i; i = 1, \dots, n\}$ is thus used for estimating H and C. The permutation entropy H is the normalized Shannon entropy of P:

$$H(P) = \frac{1}{\ln(n)} \sum_{i=0}^{n} p_i \ln(1/p_i), \quad \text{(A1)}$$

where $\ln(n)$ corresponds to the maximum value of the Shannon entropy $S(P) = \sum_{i=0}^{n} p_i \ln(1/p_i)$, occurring when all permutations are equally likely to occur ($p_i = 1/n$). While the statistical complexity is defined by

$$C(P) = \frac{D(P, U)H(P)}{D^*}, \tag{A2}$$

in which

$$D(P, U) = S\left(\frac{P + U}{2}\right) - \frac{S(P)}{2} - \frac{S(U)}{2} \tag{A3}$$

is the Jensen-Shannon divergence between P and the uniform distribution $U = \{u_i = 1/n; i = 1, \dots, n\}$ and D^* is a normalization constant [obtained by calculating $D(P, U)$ when $P = \{p_i = \delta_{1,i}; i = 1, \dots, n\}$].

The entropy H is a measure of "disorder" in the occurrence order of the elements of A. Values of $H \approx 1$ indicate that these elements appear in random order, while values of $H \approx 0$ imply that they tend to appear in a certain order. On the other hand, the values of C quantify the "structural" complexity present in the matrix A. For a given value of H, the complexity C can assume values between a minimum and a maximum and provides important additional information about the correlational structure of A that is not properly carried out by the values of H. Mainly for this reason, we have used the diagram of C versus H (the so-called complexity-entropy plane) as a discriminating tool for investigating the liquid crystal textures. This framework has been successfully used in several applications with time series [34–38] and image analysis [29,39–41]. In addition to their simplicity and intuitive meaning, these complexity measures are very fast and scalable from the computational point of view. This approach has also only the embedding dimensions d_x and d_y as "tuning parameters." However, this choice is not completely arbitrary, and the condition $(d_x d_y)! \ll n_x n_y$ must hold in order to obtain a reliable estimation of P. We have used $d_x = d_y = 2$ in the study of simulated nematic textures and $d_x = 2$ and $d_y = 3$ in all other applications due to the dimensions of the matrices associated with the textures. However, very similar results are obtained when considering $d_x = 3$ and $d_y = 2$, $d_x = 2$ and $d_y = 3$, or $d_x = d_y = 2$ in all applications.

APPENDIX B: MONTE CARLO SIMULATIONS

In order to obtain the nematic textures analyzed in Sec. II A, we have simulated a system composed of headless spins located over the sites of a tridimensional cubic lattice of dimensions $N_x \times N_y \times N_z$ (with $N_x = N_y = 100$ and $N_z = 20$). These spins have directions represented by unit vectors $\vec{u}_i$ [$i = (1, 2, \dots, N)$, with $N = N_x N_y N_z = 200,000$]. The spins in the first layer of the z direction are fixed and point to the y direction, while those in the last layer are fixed along the x direction. These two layers of fixed spins mimic the surface region (denoted by $\mathcal{S}$) and supply an anchoring direction that twists the alignment of the spins across the sample. The other spins in the bulk region (denoted by $\mathcal{B}$) interact with their nearest neighbors via the Lebwohl-Lasher potential [42] with periodic boundary conditions along the x and y directions. The Hamiltonian of this system can be written as

$$U_{\mathrm{N}} = \frac{1}{2} \sum_{\substack{i,j \in \mathcal{B} \\ i \neq j}} \Phi_{ij} + J \sum_{\substack{i \in \mathcal{B} \\ j \in \mathcal{S}}} \Phi_{ij}, \tag{B1}$$

in which J is the strength of the anchoring energy and

$$\Phi_{ij} = -\epsilon_{ij}\left[\tfrac{3}{2}\cos(\vec{u}_i \cdot \vec{u}_j) - \tfrac{1}{2}\right], \tag{B2}$$

with $\epsilon_{ij} = \epsilon$ when i and j are nearest neighbors, and zero otherwise.

All textures are obtained with $J = 1$, and the bulk spins are initially aligned making an angle with respect to the x direction [$u_i = (\cos(0.3), \sin(0.3), 0)$] for avoiding the appearance of unstable defects [43,44]. The lattice updates are obtained through the Metropolis algorithm. Each new configuration is generated by following the Barker-Watts technique [45] and is accepted with probability $\exp(-\frac{\Delta U}{k_B T})$, where ΔU is the energy difference between the old and new states, T the temperature, and k_B the Boltzmann constant. A Monte Carlo step is completed when all spins are updated on average.

The simulations start with a given reduced temperature $T_r = k_B T/\epsilon$, and the system is initially simulated for 10^4 Monte Carlo steps to avoid transient behaviors. Next, we consider another 10^4 Monte Carlo steps for estimating the average order parameter over each layer of the system. The local order parameter is the largest eigenvalue of the order matrix

$$Q_{ab} = \langle u_i^{(a)} u_i^{(b)} - \delta_{ab} \rangle, \tag{B3}$$

where $u_i^{(a)}$ and $u_i^{(b)}$ are the ath and bth components of the unitary vector $\vec{u}_i$ associated with the ith spin, and δ_{ab} stands for the Kronecker delta. The order parameter across the entire sample (p) is calculated by averaging the mean value of each free layer of the system. This model is well known to present a bulk phase transition at the critical temperature $T_c = 1.1232$ [46], that decreases when considering a confined sample with hybrid boundary conditions [47]. In our case, we have found that the critical temperature is $T_c = 1.1075$. For simulations with reduced temperatures closer to this critical temperature, the system is initially simulated for 9×10^4 Monte Carlo steps for avoiding transient behaviors related to the phase transition. The textures are obtained by averaging the latest 50 Monte Carlo steps via the Stokes-Muller methodology [48]. This procedure consists in treating incoming light (parallel to the z direction) as a Stokes vector, and describing each site as a Muller matrix. We have considered $n_e = 1.66$ for the extraordinary refraction index, $n_0 = 1.50$ for the ordinary refraction index, a sample thickness of 5.3 μm, and wavelength of 545 nm for the incoming light. The resulting texture is represented by a matrix with dimensions 100 × 100, whose elements stand for the transmitted light intensity around a particular site of the surface area.

APPENDIX C: EXPERIMENTAL PROCEEDINGS AND IMAGE FILES PROCESSING

The experimental textures analyzed in Sec. II B are obtained via polarized optical microscope imaging of liquid crystal samples at different temperatures. The samples consist of rectangular capillaries with no surface treatment

(300 μm × 4 mm) filled with the E7 mixture at 70 °C to avoid flow alignment. Next, the samples are cooled up to room temperature and placed on a temperature controller under the polarized optical microscope setup. We start taking pictures of textures from samples at 40 °C. The samples are slowly heated at a constant rate of 0.2 °C per minute, and pictures are taken every 90 s until the temperature reaches 55 °C. For higher temperatures, the heating rate is reduced to 0.05 °C per minute, and pictures are taken every 60 s until the temperature of 61 °C is achieved.

All acquired image files are in PNG format with dimensions of 2047 pixels width by 1532 pixels height, and 24 bits per pixel (8 bits for each one of the three layers in the RGB "color space"). This means that each pixel has 256 possible intensities of red, green, and blue colors, allowing more than 16 million color variations. These files can be represented by a three-layer matrix of dimensions n_x (image width) by n_y (image height), in which each layer corresponds to a color channel whose elements are the color intensities (ranging from 0 to 255). We have calculated $0.2125R + 0.7154G + 0.0721B$, that is, a weighted-average over the three layers, where R, G and B stand for the shade intensities of red, green, and blue colors of each pixel. This procedure corresponds to the grayscale luminance (or reflectance) transformation [49], which is considered to mimic the color sensibility of the human eye. This procedure yields a single matrix for each image file from which the values of H and C are calculated.

APPENDIX D: SIMULATIONS OF THE CONTINUUM ELASTIC THEORY

The cholesteric textures investigated in Sec. II C are obtained via the continuum elastic theory. In particular, we have used the Landau-de Gennes approach [32] for describing the energy density F associated with variations in the tensorial order parameter Q around the equilibrium state. By letting x_1, x_2, and x_3 represent the spatial coordinates, the energy density can be written as

$$F = \frac{L_1}{2}\frac{\partial Q_{ij}}{\partial x_k}\frac{\partial Q_{ij}}{\partial x_k} + \frac{L_2}{2}\frac{\partial Q_{ij}}{\partial x_j}\frac{\partial Q_{ik}}{\partial x_k} + \frac{L_3}{2}Q_{ij}\frac{\partial Q_{kl}}{\partial x_i}\frac{\partial Q_{kl}}{\partial x_j} + \frac{4\pi}{\eta}L_q\epsilon_{ikl}Q_{ij}\frac{\partial Q_{lj}}{\partial x_k} + \frac{A}{2}Q_{ij}Q_{ji} + \frac{B}{3}Q_{ij}Q_{jk}Q_{ki} + \frac{C}{4}Q_{ij}Q_{jk}Q_{kl}Q_{li}, \tag{D1}$$

where L_1, L_2, L_3, and L_q are elastic constants, A, B, and C are thermodynamic parameters, and η is the cholesteric pitch length. Here, we have assumed implicit summation in repeated indexes. The time evolution of the components of Q_{ij} is given by

$$\Gamma\frac{\partial Q_{ij}}{\partial t} = \left[\frac{\partial F(Q)}{\partial Q_{ij}} - \frac{d}{dx_k}\frac{\partial F(Q)}{\partial Q_{ij,k}}\right], \tag{D2}$$

where Γ is the liquid crystal rotational viscosity, t is the time and $Q_{ij,k}$ is the derivative of Q_{ij} relative to x_k.

The system of equations (D2) is numerically solved via finite differences method in a uniform grid with 200 × 200 × 20 grid points. All distance units are normalized by the grid distance $\delta x_1 = 1$ nm, $\delta x_2 = 1$ nm and $\delta x_3 = 1$ nm. The liquid crystal parameters used are $A = -0.348$ MJ/(Km^3), $B = -2.133\,\text{MJ/m}^3$, $C = 1.733\,\text{MJ/m}^3$, $\Gamma = 0.3$ Pa s, $L_1 = 2.6$ pN, $L_2 = 2.6$ pN, $L_3 = 0.76$ pN, and $L_q = 1.86$ pN. These parameters are obtained from the literature and are well-known to roughly describe the 5CB liquid crystal [3]. Thus, instead of rescaling the elastic and thermodynamic constants to unrealistic values [32], we have used real values for these parameters but a small lattice size (δx_i). This choice implies in using small and unrealistic values for the cholesteric pitches. However, the textures produced with this approach are qualitatively very similar to those obtained when using unrealistic values for the physical parameters. Also, these textures are well known to mimic well those obtained from experimental studies. We have considered different values for the pitch η and the Dormand-Prince of fourth-order [50] for the time integration. The initial condition is randomly chosen from a uniform distribution and periodic boundary conditions are considered in all directions for avoiding surface effects. Finally, the resulting optical textures are generated by applying the Jones 2 × 2 method [30] with the liquid crystal cells placed between crossed polarizers. This approach is the same used in Ref. [51] for generating the optical images.

APPENDIX E: IMPLEMENTATION OF THE MACHINE LEARNING ALGORITHMS

Machine learning tasks include classification and regression. Classification is the task of predicting a discrete class label by providing a correctly labeled training set of data, whereas in regression tasks the algorithm predicts a continuous quantity. Thus, the predictions of the order parameter from the simulated nematic textures and the predictions of the temperature from the experimental nematic textures represent a regression task. On the other hand, the predictions of the discrete set of cholesteric pitches represent a classification task. For both tasks, we have used the k-nearest neighbors algorithms as implemented in the Python module scikit-learn [52]. This is one of the simplest machine learning algorithms that make predictions based on the classes (or values) of k-closest neighbors [15]. The only parameter of this algorithm is the number of nearest neighbors k.

The best value of k is chosen to simultaneously minimize the bias and variance errors of the predictions (the so-called bias-variance tradeoff [15]). Bias errors occurs when the statistical learning algorithm is not complex enough to capture the underlying structure of the data (underfitting). On the other hand, variance errors appear when the algorithm is too complex and starts to model even the noise in the training set, but fail in predicting future observations in unseen data (overfitting). There is thus a trade-off between minimizing the bias and variance errors concerning the algorithm complexity.

To address this question, we employ a resampling strategy known as n-fold cross-validation [15]. This approach consists in randomly splitting the data into n subsamples of approximately equal size. One of the subsamples is separated for validating the algorithm, and the remaining $n - 1$ are used for training the algorithm. This process is thus repeated n times. The accuracy obtained from the training set is the training score, and the one obtained from the validation set is the cross-validation score. At the end of n repetitions, we have n estimates for the training and cross-validation scores, from which we calculate average values and confidence intervals. The plot of the training and cross-validation scores as a function of model parameters (in our case, the number of nearest neighbors k) is known as validation curves. In our regression tasks, the scores represent the coefficient of determination (R^2) of the relationship between the predicted and true values. When positive, the R^2 is also interpreted as the percentage (or fraction) of the response variable variation that is explained by the model. Also, in our classification task, the score (or accuracy) represents the fraction of correct classifications. An underfitting situation happens when both training and validation scores are low; whereas overfitting is represented by good training scores and poor validation scores. The best bias-variance tradeoff occurs when the highest values are obtained for both scores. We have also estimated the learning curves, which represent the dependence of the training and the cross-validation scores on the size of the training set. This is an important matter for statistical learning methods because small training sets may not be enough for correctly fitting the model, while unnecessary data may introduce noise to the model.

We have further compared the performance of the k-nearest neighbors algorithm with other more complex machine learning methods (namely: random forest, support vector machine, and neural network). However, the scores for these methods are very similar to those obtained for the k-nearest neighbors. For this reason and also due to the simplicity of the k-nearest neighbors, we have only reported the results for this machine learning method.

[1] F. Chen, G. M. Brown, and M. Song, Opt. Eng. **39**, 10 (2000).

[2] Z. Wang, H. Li, Y. Zhu, and T. Xu, Arch. Comput. Methods Eng. **24**, 637 (2017).

[3] J. Prost and P. G. de Gennes, *The Physics of Liquid Crystals*, Vol. 83 (Oxford University Press, Oxford, 1995).

[4] R. S. Zola, L. R. Evangelista, Y.-C. Yang, and D.-K. Yang, Phys. Rev. Lett. **110**, 057801 (2013).

[5] M. Shribak and R. Oldenbourg, Appl. Opt. **42**, 3009 (2003).

[6] R. Oldenbourg, in *Live Cell Imaging: A Laboratory Manual* (Cold Spring Harbor Laboratory Press, New York, 2005), pp. 205–237.

[7] S. Zhou, S. V. Shiyanovskii, H.-S. Park, and O. D. Lavrentovich, Nat. Commun. **8**, 14974 (2017).

[8] S. J. DeCamp, G. S. Redner, A. Baskaran, M. F. Hagan, and Z. Dogic, Nat. Mater. **14**, 1110 (2015).

[9] B. Montrucchio, A. Sparavigna, and A. Strigazzi, Liq. Cryst. **24**, 841 (1998).

[10] S. S. Sastry, K. Mallika, B. G. S. Rao, H. S. Tiong, and S. Lakshminarayana, Liq. Cryst. **39**, 415 (2012).

[11] S. S. Sastry, S. Kumar, T. V. Kumari, K. Mallika, B. G. S. Rao, and H. S. Tiong, Liq. Cryst. **39**, 1527 (2012).

[12] S. S. Sastry, B. G. S. Rao, K. B. Mahalakshmi, K. Mallika, C. N. Rao, and H. S. Tiong, ISRN Condens. Matter Phys. **2012**, 423650 (2012).

[13] Z.-g. Zheng, R. S. Zola, H. K. Bisoyi, L. Wang, Y. Li, T. J. Bunning, and Q. Li, Adv. Mater. **29**, 1701903 (2017).

[14] H. Nemati, D.-K. Yang, K.-L. Cheng, C.-C. Liang, J.-W. Shiu, C.-C. Tsai, and R. Zola, J. Appl. Phys. **112**, 124513 (2012).

[15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics (Springer, New York, 2013).

[16] L.-F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, and A. J. Millis, Phys. Rev. B **90**, 155136 (2014).

[17] A. G. Kusne, T. Gao, A. Mehta, L. Ke, M. C. Nguyen, K.-M. Ho, V. Antropov, C.-Z. Wang, M. J. Kramer, C. Long *et al.*, Sci. Rep. **4**, 6367 (2014).

[18] E. D. Cubuk, S. S. Schoenholz, J. M. Rieser, B. D. Malone, J. Rottler, D. J. Durian, E. Kaxiras, and A. J. Liu, Phys. Rev. Lett. **114**, 108001 (2015).

[19] S. V. Kalinin, B. G. Sumpter, and R. K. Archibald, Nat. Mater. **14**, 973 (2015).

[20] J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017).

[21] Q. Wei, R. G. Melko, and J. Z. Y. Chen, Phys. Rev. E **95**, 032504 (2017).

[22] W. Hu, R. R. P. Singh, and R. T. Scalettar, Phys. Rev. E **95**, 062122 (2017).

[23] C. Dietz, T. Kretz, and M. H. Thoma, Phys. Rev. E **96**, 011301 (2017).

[24] J. Venderley, V. Khemani, and E.-A. Kim, Phys. Rev. Lett. **120**, 257204 (2018).

[25] C. Bandt and B. Pompe, Phys. Rev. Lett. **88**, 174102 (2002).

[26] O. A. Rosso, H. A. Larrondo, M. T. Martin, A. Plastino, and M. A. Fuentes, Phys. Rev. Lett. **99**, 154102 (2007).

[27] H. V. Ribeiro, L. Zunino, E. K. Lenzi, P. A. Santoro, and R. S. Mendes, PLoS ONE **7**, e40689 (2012).

[28] L. Zunino and H. V. Ribeiro, Chaos Solitons Fractals **91**, 679 (2016).

[29] H. Y. D. Sigaki, M. Perc, and H. V. Ribeiro, Proc. Natl. Acad. Sci. USA **115**, E8585 (2018).

[30] S. Wu and D. Yang, *Fundamentals of Liquid Crystal Devices*, Wiley Series in Display Technology (Wiley, Chichester, 2006).

[31] T. W. Stinson and J. D. Litster, Phys. Rev. Lett. **25**, 503 (1970).

[32] M. Ravnik and S. Žumer, Liq. Cryst. **36**, 1201 (2009).

[33] R. López-Ruiz, H. L. Mancini, and X. Calbet, Phys. Lett. A **209**, 321 (1995).

[34] T. Jovanovic, S. García, H. Gall, and A. Mejía, Stoch. Env. Res. Risk Assess. **31**, 2107 (2017).

[35] Q. Li and Z. Fu, Phys. Rev. E **89**, 012905 (2014).

[36] T. Stosic, L. Telesca, D. V. de Souza Ferreira, and B. Stosic, J. Hydrol. **540**, 1136 (2016).

[37] H. V. Ribeiro, M. Jauregui, L. Zunino, and E. K. Lenzi, Phys. Rev. E **95**, 062106 (2017).

[38] H. V. Ribeiro, L. Zunino, R. S. Mendes, and E. K. Lenzi, Physica A **391**, 2421 (2012).

[39] A. Schlemmer, S. Berg, T. Shajahan, S. Luther, and U. Parlitz, in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE* (IEEE, Milan, 2015), pp. 4049–4052.

[40] A. P. Antonelli, G. J. Meschino, and V. L. Ballarin, in *Information Processing and Control (RPIC), 2017 XVII Workshop on* (IEEE, Mar de Plata, 2017), pp. 1–7.

[41] A. Antonelli, G. Meschino, and V. Ballarin, in *World Congress on Medical Physics and Biomedical Engineering 2018* (Springer, Prague, 2018), pp. 135–141.

[42] P. A. Lebwohl and G. Lasher, Phys. Rev. A **6**, 426 (1972).

[43] C. Chiccoli, L. Evangelista, P. Pasini, R. Teixeira de Souza, and C. Zannoni, Mol. Cryst. Liq. Cryst. **614**, 137 (2015).

[44] C. Chiccoli, L. Evangelista, E. Omori, P. Pasini, R. Teixeira-Souza, and C. Zannoni, Mol. Cryst. Liq. Cryst. **649**, 86 (2017).

[45] J. Barker and R. Watts, Chem. Phys. Lett. **3**, 144 (1969).

[46] U. Fabbri and C. Zannoni, Mol. Phys. **58**, 763 (1986).

[47] C. Chiccoli, P. Pasini, A. Sarlah, C. Zannoni, and S. Žumer, Phys. Rev. E **67**, 050703 (2003).

[48] E. Berggren, C. Zannoni, C. Chiccoli, P. Pasini, and F. Semeria, Phys. Rev. E **50**, 2929 (1994).

[49] S. van der Walt, J. L. Schönberger, J. Nunez Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, the scikit-image contributors, PeerJ **2**, e453 (2014).

[50] J. R. Dormand and P. J. Prince, J. Comput. Appl. Math. **6**, 19 (1980).

[51] D. Seč, T. Porenta, M. Ravnik, and S. Žumer, Soft Matter **8**, 11982 (2012).

[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, J. Machine Learn. Res. **12**, 2825 (2011).