

## Accelerated Search and Design of Stretchable Graphene Kirigami Using Machine Learning

Paul Z. Hanakata,<sup>1,\*</sup> Ekin D. Cubuk,<sup>2</sup> David K. Campbell,<sup>1</sup> and Harold S. Park<sup>3</sup>

<sup>1</sup>*Department of Physics, Boston University, Boston, Massachusetts 02215, USA*

<sup>2</sup>*Google Brain, Mountain View, California 94043, USA*

<sup>3</sup>*Department of Mechanical Engineering, Boston University, Boston, Massachusetts 02215, USA*



(Received 14 August 2018; published 20 December 2018)

Making kirigami-inspired cuts into a sheet has been shown to be an effective way of designing stretchable materials with metamorphic properties where the 2D shape can transform into complex 3D shapes. However, finding the optimal solutions is not straightforward as the number of possible cutting patterns grows exponentially with system size. Here, we report on how machine learning (ML) can be used to approximate the target properties, such as yield stress and yield strain, as a function of cutting pattern. Our approach enables the rapid discovery of kirigami designs that yield extreme stretchability as verified by molecular dynamics (MD) simulations. We find that convolutional neural networks, commonly used for classification in vision tasks, can be applied for regression to achieve an accuracy close to the precision of the MD simulations. This approach can then be used to search for optimal designs that maximize elastic stretchability with only 1000 training samples in a large design space of  $\sim 4 \times 10^6$  candidate designs. This example demonstrates the power and potential of ML in finding optimal kirigami designs at a fraction of iterations that would be required of a purely MD or experiment-based approach, where no prior knowledge of the governing physics is known or available.

DOI: [10.1103/PhysRevLett.121.255304](https://doi.org/10.1103/PhysRevLett.121.255304)

*Introduction.*—Recently, there has been significant interest in designing flat sheets with metamaterial-type properties, which rely upon the transformation of the original 2D sheet into a complex 3D shape. These complex designs are often achieved by folding the sheet, called the origami approach, or by patterning the sheet with cuts, called the kirigami approach. Owing to the metamorphic nature, designs based on origami and kirigami have been used for many applications across length scales, ranging from meter-size deployable space satellite structures [1] to soft actuator crawling robots [2] and micrometer-size stretchable electronics [3,4].

Atomically thin two-dimensional (2D) materials such as graphene and MoS<sub>2</sub> have been studied extensively due to their exceptional physical properties (mechanical strength, electrical conductivity, etc). Based on experiments [4] and atomistic simulations [5,6], it has been shown that introducing arrays of kirigami cuts allows graphene and MoS<sub>2</sub> to buckle in the direction perpendicular to the plane. These out-of-plane buckling and rotational deformations are key to enabling significant increases in stretchability.

By the principles of mechanics of springs, it is expected that adding cuts (removing atoms) generally will both soften and weaken the material. Griffith's criterion for fracture [7] has been successfully used to explain the decrease in fracture strength for a single cut [8–11], but cannot explain how the delay of failure is connected to the out-of-plane deflection of kirigami cuts. Several analytical

solutions have been developed to explain the buckling mechanism in single cut geometries [12,13], a square array of mutually orthogonal cuts [14], and a square hole [15]. These analytical solutions are applicable for regular repeating cuts, but may not be generally applicable for situations where nonuniform and nonsymmetric cuts may enable superior performance.

An important, but unresolved question with regards to kirigami structures at all length scales is how to locate the cuts to achieve a specific performance metric. This problem is challenging to solve due to the large numbers of possible cut configurations that must be explored. For example, the typical size scale of current electronic devices is micrometers ( $10^{-6}$  meters) and the smallest cuts in current 2D experiments are about  $10 \times 10$  Å [16]. Thus, exhaustively searching for good solutions in this design space would be impractical as the number of possible configurations grows exponentially with the system size. Alternatively, various optimization algorithms, i.e., genetic and greedy algorithms, and topology optimization approaches, have been used to find optimal designs of materials based on finite element methods [17–20]. However, these approaches have difficulties as the number of degrees of freedom in the problem increases, and also if the property of interest lies within the regime of nonlinear material behavior.

Machine learning (ML) methods represent an alternative, and recently emerging approach to designing materials where the design space is extremely large. For example,

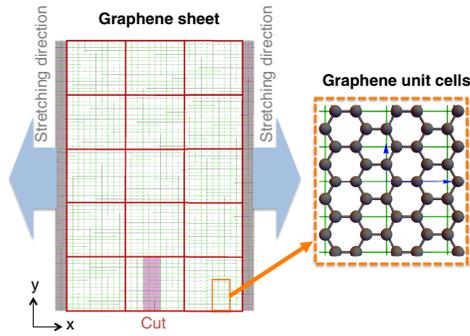


FIG. 1. Schematic diagrams of a graphene sheet and rectangular graphene unit cells. Each of the grid (colored red) consists of  $10 \times 16$  rectangular graphene unit cells (colored green).

ML has been used to design materials with low thermal conductivity [21], battery materials [22,23], and composite materials with stiff and soft components [24]. ML methods have also recently been used to study condensed matter systems with quantum mechanical interactions [25–27], disordered atomic configurations [28–30], and phase transitions [31,32]. While ML is now being widely used to predict properties of new materials, there have been relatively few demonstrations of using ML to design functional materials and structures [33].

In this Letter, we use ML to systematically study how the cut density and the locations of the cuts govern the mechanical properties of graphene kirigami. We use fully connected neural networks (NNs) and also convolutional neural networks (CNNs) to approximate the yield strain and stress. To formulate this problem systematically, we partition the graphene sheets into grids, where atoms in each grid region will either be present or cut, as shown schematically in Fig. 1. We then utilize the CNNs for inverse design, where the objective is to maximize the elastic stretchability of the graphene kirigami subject to a constraint on the number of cuts. We use ML to search through a design space of approximately 4 000 000 possible configurations, where it is not feasible to simulate all possible configurations in a brute force fashion. Despite the size of the design space, our model is able to find the optimal solution with fewer than 1000 training data points [evaluations via molecular dynamics (MD)]. Our findings can be used as a general method to design a material without any prior knowledge of the fundamental physics, which is particularly important for designing materials when only experimental data are available and an accurate physical model is unknown.

*Overview of mechanical properties.*—In this section, we give a brief overview of the changes in the mechanical properties of graphene with cuts. The 2D binary array of cut configurations  $N_{\text{grid}} = N_x \times N_y$  is flattened into a one-dimensional array vector  $\mathbf{x}$  of size  $n = N_{\text{grid}}$ . We use  $n$  for number of features,  $m$  for the number of samples,  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$  for the binary vector describing cut

configurations,  $\vec{x}, \vec{y}, \vec{z}$  for the real space vectors (atomic locations), and  $\hat{x}, \hat{y}, \hat{z}$  for the unit vectors in real space.

We study one unit kirigami of size  $\sim 100 \times 200 \text{ \AA}$ , where cuts are allowed to be present on the  $3 \times 5$  grid; this gives a design space of  $2^{15} = 32768$  possible cut configurations (Fig. 1). Each cell of the grid also consists of  $10 \times 16$  rectangular graphene unit cells. There are 2400 rectangular graphene unit cells in this sheet; there are four carbon atoms in the rectangular graphene unit cell. This gives a total of 9600 carbon atoms in a kirigami sheet without cuts. In this system, the cut density can range from 0 cuts in the 15 grids to 15 cuts in the 15 grids, while keeping each cut size constant at  $12 \times 38 \text{ \AA}$  ( $3 \times 16$  rectangular graphene unit cells), which is relevant to current experimental capabilities [16]. Following previous work [5,12], we use the Sandia open-source MD simulation code LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [34] to generate the ground truth data for our training model, where we simulate graphene as the 2D constituent material of choice for the kirigami at a low temperature of 4.2 K. Since we simulate MD at  $T = 4.2 \text{ K}$ , the obtained yield strain (or stress) of a configuration varies due to stochasticity (i.e., distributions of the initial velocities). The MD precisions for strain and stress are  $\eta^\epsilon = 0.046$  and  $\eta^\sigma = 2.00 \text{ GPa}$ , respectively. In this work, we focus only on kirigami with armchair edges along the  $\hat{x}$  direction as the stretchability is improved regardless of the chirality of graphene with armchair or zigzag edges [5]. The sheets are stretched in the  $\hat{x}$  direction and engineering strain  $\epsilon = L/L_0 - 1$  is used to quantify stretchability, where  $L_0$  and  $L$  are the length of sheet in the direction of the loading before and after the deformation, respectively. More details of simulations can be found in the Supplemental Material (SM) [35].

Stress-strain curves of three representative cuts are shown in Fig. 2(a). From the stress-strain curve we can identify the ultimate stress  $\sigma^u$  (and the corresponding strain  $\epsilon^u$ ), yield stress  $\sigma^y$ , and yield strain  $\epsilon^y$ . Another point of interest is the failure point where the system is completely detached upon stretching. Here, we focus on the yield point where plastic deformation or bond breaking begins. Analyzing the amount of strain between yielding and the onset of failure would give insights into the postyielding deformation mechanisms, and the role of kirigami structure in controlling the postyield behavior, which could be done in a subsequent study. As shown in Fig. 2(b), the  $\sigma^y$  consistently decreases with increasing number of cuts.  $\epsilon^y$  has much more variability at higher cut density. At a cut density of 73% (11 cuts),  $\epsilon^y$  varies over a wide range of values from  $\sim 0.2$  (20%) to  $\sim 2.0$  (200%). This shows that increasing the number of cuts without intelligently locating the cuts may not always increase the stretchability.

*Machine learning.*—We train NNs and CNNs to predict the yield strain in the context of supervised learning. Two-dimensional images of size  $30 \times 80$  are used as inputs for training the CNNs. For the NNs, the 2D images are

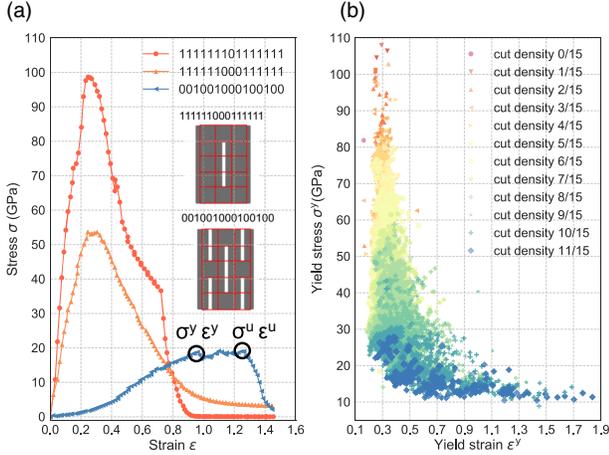


FIG. 2. (a) Stress-strain plot of three representative kirigamis. Inset shows the “typical” kirigami cuts. (b) Yield stress as a function of yield strain for different configurations. Data are colored based on their cut density.

flattened to 1D arrays of size 2400. The 2400 grids correspond to the number of rectangular graphene unit cells. In vision tasks the CNN is usually used for classification. Here, we will use both the NN and CNN for *regression*. Accordingly, we do not include the activation function at the end of the final layer, and we minimize the mean squared error loss to optimize the model parameters.

Since the yield strain and yield stress results are similar as they are roughly inversely proportional to each other [see Fig. 2(b)], we will focus on the yield strain. All plots and data for yield stress can be found in the SM. Out of  $2^{15}$  possible configurations, only the 29 791 nondetached configurations are considered. We find the nondetached configurations, where nondetached means that there is no cut that traverses the entire length (in the  $y$  direction) of the kirigami. We randomly shuffle the 29 791 data samples, and use 80% for training, 10% for validation, and 10% for the test set. [42]. The validation data set is used to find better architectures (“hyperparameter tuning,” i.e., changing number of neurons or filters), and the test data set is used to assess performance. We provide details on the hyperparameters and the performance of different CNN and NN architectures in the SM.

We use simple shallow NNs with one hidden layer of size ranging from 4 and 2024. For a CNN, we use architectures similar to VGGNET [43]. The kernel size is fixed at  $3 \times 3$  with a stride of 1. Each convolutional layer is followed by a rectified linear unit function and a max-pooling layer of size  $2 \times 2$  with stride of 2 [44]. Based on validation data set performance, here we report the best performing CNN and NN architecture.

We use the root-mean-square error (RMSE) and  $R^2$  on the test set to evaluate the goodness of a model. A CNN with number of filters of 16, 32, 64 in the first, second, and third convolutional layer, respectively, and a fully

connected layer (FCL) of size 64 achieves  $R^2 = 0.92$  and RMSE of 0.053 which is close to the MD precision of 0.046. We will denote this CNN model by CNN- $f16-f32-f64-h64$ ; here “ $f$ ” stands for filter and “ $h$ ” stands for number of neurons in the FCL. A NN with 64 neurons achieves  $R^2 = 0.84$  and a RMSE of 0.075. A NN with 246 neurons achieves a RMSE of 0.123 and a CNN with 256 FCL achieves a RMSE of 0.054. We found that increasing the number of neurons in a NN does not improve the accuracy. In addition, we use simple ordinary least square (OLS) regression to see how the CNN performs compared to such a simpler model. For yield strain, a polynomial degree of 3 gives  $R^2 = 0.76$  and RMSE = 0.084. The CNN performs better than the NN and OLS as the CNN learns from the local 2D patterns. Performance of the CNN and NN with different architectures (different neurons number ranging from 4 to 2024) as well as simple OLS, and details on the RMSE, MD precision and  $R^2$  can be found in the SM.

*Inverse design of highly stretchable kirigami.*—In the previous section, we used a NN and CNN for the prediction of mechanical properties, in the context of supervised learning. Next, we investigate if the approximated function can be used to search for optimal designs. Here, we will use a CNN, the best performing model, to search for the cut configuration with the largest yield strain. The procedure is as follows: first we randomly choose 100 configurations from the library of all possible configurations and use MD to obtain the yield strain. After training, the CNN then is used to screen the *unexplored* data set for the top performing 100 remaining candidates. Based on this screening, 100 new MD simulations are performed and the results are added to the training set for the next generation. The ML search algorithm flow diagram is shown in Fig. 3(a). The difference from the previous section is that here we train the model incrementally with the predicted top performers.

We first use the  $3 \times 5$  allowed cuts where we already have simulated *all* of the possible configurations in MD to make sure that our model indeed finds the *true* (or close to) optimal designs. To evaluate the performance of the search algorithm we use the average of yield strain of the top 100 performers  $\bar{\epsilon}_{\text{top } 100}^y$  for each generation. This number, which cannot be too small, is chosen arbitrarily so that we obtain more than a handful of good candidates. As a benchmark, we include the “naive” random search. Specifically, we use CNN- $f16-f32-f64-h64$  architecture to find the optimal designs. As shown in Fig. 3(b), the random search needs 30 generations (3000 MD simulations) to get  $\bar{\epsilon}_{\text{top } 100}^y \geq 1.0$  (100% strain) and explore the entire design space in order to find the true best 100 performers. The CNN approach requires only 3 generations (300 MD simulations data) to search for 100 candidates with  $\bar{\epsilon}_{\text{top } 100}^y \sim 1.0$  and 10 generations to search the *true* top 100 performers. In each generation the standard deviation of  $\bar{\epsilon}_{\text{top } 100}^y$  is around 0.25. Using the CNN to search for

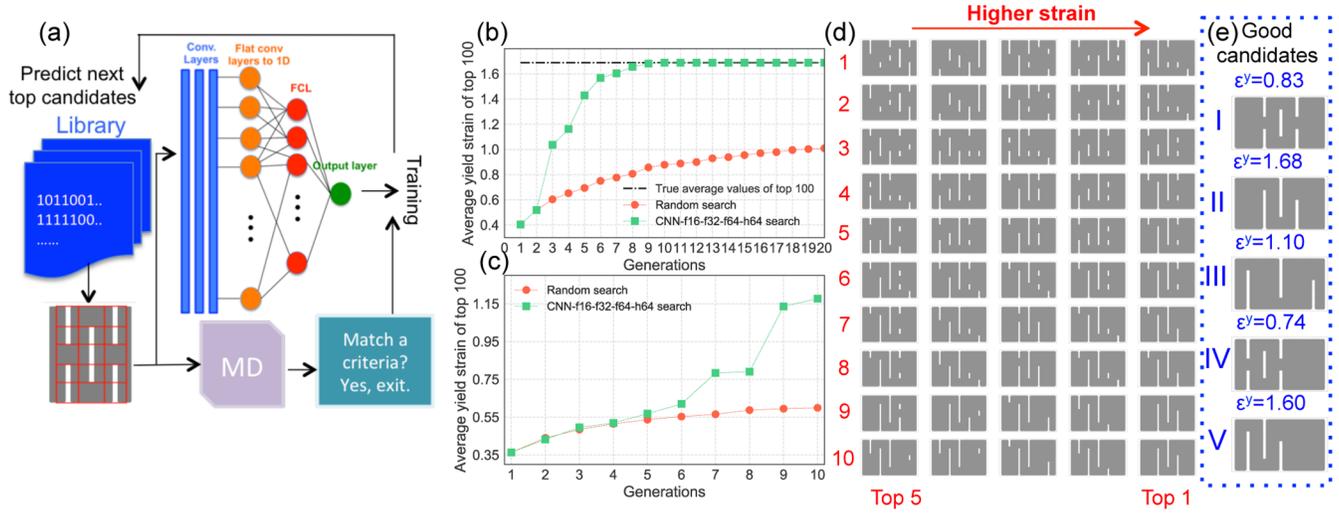


FIG. 3. (a) Schematic of the neural network search algorithm. Average yield strains of the top 100 performers as a function of generations for kirigami with allowed cuts of (b)  $3 \times 5$  and (c)  $5 \times 5$ . (d) Visualization of the top five performers of kirigami with  $5 \times 5$  allowed cuts in each generation. After the ninth generation, the top three performers remain constant. (e) A comparison between the top performing configurations found by the ML and the typical kirigami configurations with centering cuts. Note that the kirigami visualizations are not scaled to the real physical dimensions for clarity.

optimal designs is crucial because one MD simulation of graphene with a size of  $100 \times 200 \text{ \AA}$  requires around 1 h computing time using four cores of the CPU. In each generation, the required time to train the CNN and to predict the yield strain of one configuration is around 6 ms on four CPU cores (same machines) or 3 ms on four CPU cores plus one GPU [45]. From Fig. 2(b), we know that sheets with high strains are ones with high cut density. However, the variability is also large; e.g., at 11/15 cut density, the yield strain ranges from 0.2 to 1.7. Despite this complexity, the ML quickly learns to find solutions at high cut density and also to find the right cutting patterns.

Next, we apply this simple algorithm to a much larger design space where the true optimal designs are *unknown* and also with a specified design constraint. Specifically, we study larger graphene sheets by extending the physical size in  $\hat{x}$  from  $\sim 100 \times 200$  to  $\sim 200 \times 200 \text{ \AA}$  (from  $30 \times 80$  to  $50 \times 80$  rectangular graphene unit cells). For this system, one MD simulation requires around 3 h of computing time running on four cores. The allowed cuts are also expanded from  $3 \times 5$  to  $5 \times 5$  grids. For this problem, we fixed the number of cuts at 11 cuts, which gives a design space of size  $(25!/11!14!) \sim 4 \times 10^6$ . For this system, we could not use brute force to simulate all configurations as we did previously for a system with 15 allowed cuts. While the typical stretchable kirigamis usually have cuts and no cuts along the loading direction ( $\hat{x}$ ), it is not clear whether all the cuts should be located closely in a region or distributed equally.

As shown in Fig. 3(b), the CNN is able to find designs with higher yield strains. With fewer than 10 generations (1000 training data), the CNN is able to find configurations with yield strains  $\geq 1.0$ , which is roughly *five* times larger

than a sheet without cuts. In each generation, the standard deviation of the top 100 performers is around 0.1. In Fig. 3(d), we plot cut configurations of the top five performers in each generation. It can be seen that the cut configurations are random in the early stage of the search but evolve quickly to configurations with a long cut along the  $\hat{y}$  direction alternating in  $\hat{x}$  direction, as we expected from the smaller grid system. This suggests that our ML approach is scalable in a sense that the same CNN architecture used previously in the simpler system with 15 allowed cuts can search the optimal designs effectively despite a large design space.

We next take a closer look on the top performing configurations. Interestingly, the optimal solutions for maximum stretchability found by a CNN have cuts at the edges which are different from the “typical” kirigami with centering cuts [Fig. 3(e) configuration I]. The found best performer has a yield strain *twice* as large as the kirigami with centering cuts. We found that to achieve high yield strains the long cuts should be located close to each other, rather than being sparsely or equally distributed across the sheet along the  $\hat{x}$  direction, as shown by comparing configurations II and III in Fig. 3(e). These overlapping cut configurations allow larger rotations and out-of-plane deflection which give higher stretchability; i.e., the alternating edge cut pattern effectively transforms the 2D membrane into a quasi-1D membrane. Close packing of the alternating edge cuts allows increased stretchability because the thinner ribbons connecting different segments improve twisting. This result is similar to what we found previously in kirigami with centering cuts [5,6]. Visualizations illustrating these effects (such as out-of-plane buckling and twisting) and a more detailed

discussion can be found in the SM. This design principle is particularly useful as recently a combination of dense and sparse cut spacing were used to design stretchable thin electronic membranes [46]. It is remarkable not only that ML can quickly find the optimal designs using few training data ( $< 1\%$  of the design space) under certain constraints, but also that ML can capture uncommon physical insights needed to produce the optimal designs, in this case related to the cut density and locations of the cuts.

*Conclusion.*—We have shown how machine learning methods can be used to design graphene kirigami, where yield strain and stress are used as the target properties. We found that the CNN with three convolutional layers followed by one fully connected layer is sufficient to find the optimal designs with relatively few training data. Our work shows not only how to use ML to effectively search for optimal designs but also to give new understanding on how kirigami cuts change the mechanical properties of graphene sheets. Furthermore, the ML method is parameter-free, in the sense that it can be used to design any material without any prior physical knowledge of the system. As the ML method only needs data, it can be applied to experimental work where the physical model is not known and cannot be simulated by MD or other simulation methods. Based on previous work indicating the scale invariance of kirigami deformation [12], the kirigami structures found here using ML should also be applicable for designing larger macroscale kirigami structures.

P. Z. H. developed the codes, performed the simulations and data analysis, and wrote the manuscript with input from all authors. P. Z. H. and E. D. C. developed the machine learning methods. P. Z. H., D. K. C. and H. S. P. acknowledge the Hariri Institute Research Incubation Grant No. 2018-02-002 and the Boston University High Performance Shared Computing Cluster. P. Z. H. is grateful for the Hariri Graduate Fellowship. P. Z. H. thank Grace Gu and Adrian Yi for helpful discussions.

\*paul.hanakata@gmail.com

- [1] S. A. Zirbel, R. J. Lang, M. W. Thomson, D. A. Sigel, P. E. Walkemeyer, B. P. Trease, S. P. Magleby, and L. L. Howell, *J Mech Des* **135**, 111005 (2013).
- [2] A. Rafsanjani, Y. Zhang, B. Liu, S. M. Rubinstein, and K. Bertoldi, *Sci. Rob.* **3**, eaar7555 (2018).
- [3] T. C. Shyu, P. F. Damasceno, P. M. Dodd, A. Lamoureux, L. Xu, M. Shlian, M. Shtein, S. C. Glotzer, and N. A. Kotov, *Nat. Mater.* **14**, 785 (2015).
- [4] M. K. Blees, A. W. Barnard, P. A. Rose, S. P. Roberts, K. L. McGill, P. Y. Huang, A. R. Ruyack, J. W. Kevek, B. Kobrin, D. A. Muller *et al.*, *Nature (London)* **524**, 204 (2015).
- [5] Z. Qi, D. K. Campbell, and H. S. Park, *Phys. Rev. B* **90**, 245437 (2014).
- [6] P. Z. Hanakata, Z. Qi, D. K. Campbell, and H. S. Park, *Nanoscale* **8**, 458 (2016).
- [7] A. A. Griffith and M. Eng, *Phil. Trans. R. Soc. A* **221**, 163 (1921).
- [8] H. Zhao and N. Aluru, *J. Appl. Phys.* **108**, 064321 (2010).
- [9] P. Zhang, L. Ma, F. Fan, Z. Zeng, C. Peng, P. E. Loya, Z. Liu, Y. Gong, J. Zhang, X. Zhang *et al.*, *Nat. Commun.* **5**, 3782 (2014).
- [10] G. Jung, Z. Qin, and M. J. Buehler, *Extreme Mech. Lett.* **2**, 52 (2015).
- [11] T. Rakib, S. Mojumder, S. Das, S. Saha, and M. Motalab, *Physica (Amsterdam)* **515B**, 67 (2017).
- [12] M. A. Dias, M. P. McCarron, D. Rayneau-Kirkhope, P. Z. Hanakata, D. K. Campbell, H. S. Park, and D. P. Holmes, *Soft Matter* **13**, 9087 (2017).
- [13] M. Isobe and K. Okumura, *Sci. Rep.* **6**, 24758 (2016).
- [14] A. Rafsanjani and K. Bertoldi, *Phys. Rev. Lett.* **118**, 084301 (2017).
- [15] M. Moshe, S. Shankar, M. J. Bowick, and D. R. Nelson, *arXiv:1801.08263*.
- [16] P. Masih Das, G. Danda, A. Cupo, W. M. Parkin, L. Liang, N. Kharche, X. Ling, S. Huang, M. S. Dresselhaus, V. Meunier *et al.*, *ACS Nano* **10**, 5687 (2016).
- [17] O. Sigmund and J. Petersson, *Struct. Optim.* **16**, 68 (1998).
- [18] M. J. Jakiela, C. Chapman, J. Duda, A. Adewuya, and K. Saitou, *Comput. Methods Appl. Mech. Eng.* **186**, 339 (2000).
- [19] X. Huang and Y. Xie, *Finite Elements in Analysis and Design* **43**, 1039 (2007).
- [20] G. X. Gu, L. Dimas, Z. Qin, and M. J. Buehler, *J. Appl. Mech.* **83**, 071006 (2016).
- [21] A. Seko, A. Togo, H. Hayashi, K. Tsuda, L. Chaput, and I. Tanaka, *Phys. Rev. Lett.* **115**, 205901 (2015).
- [22] A. D. Sendek, Q. Yang, E. D. Cubuk, K.-A. N. Duerloo, Y. Cui, and E. J. Reed, *Energy Environ. Sci.* **10**, 306 (2017).
- [23] B. Onat, E. D. Cubuk, B. D. Malone, and E. Kaxiras, *Phys. Rev. B* **97**, 094106 (2018).
- [24] G. X. Gu, C.-T. Chen, and M. J. Buehler, *Extreme Mech. Lett.* **18**, 19 (2018).
- [25] J. Behler and M. Parrinello, *Phys. Rev. Lett.* **98**, 146401 (2007).
- [26] J. Behler, *J. Chem. Phys.* **134**, 074106 (2011).
- [27] E. D. Cubuk, B. D. Malone, B. Onat, A. Waterland, and E. Kaxiras, *J. Chem. Phys.* **147**, 024104 (2017).
- [28] E. D. Cubuk, S. S. Schoenholz, E. Kaxiras, and A. J. Liu, *J. Phys. Chem. B* **120**, 6139 (2016).
- [29] S. S. Schoenholz, E. D. Cubuk, E. Kaxiras, and A. J. Liu, *Proc. Natl. Acad. Sci. U.S.A.* **114**, 263 (2017).
- [30] E. Cubuk, R. Ivancic, S. Schoenholz, D. Strickland, A. Basu, Z. Davidson, J. Fontaine, J. Hor, Y.-R. Huang, Y. Jiang *et al.*, *Science* **358**, 1033 (2017).
- [31] J. Carrasquilla and R. G. Melko, *Nat. Phys.* **13**, 431 (2017).
- [32] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, *Sci. Rep.* **7**, 8823 (2017).
- [33] G. X. Gu, C.-T. Chen, D. J. Richmond, and M. J. Buehler, *Mater. Horiz.* **5**, 939 (2018).
- [34] LAMMPS, <http://lammmps.sandia.gov>.
- [35] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevLett.121.255304> for details of simulations, machine learning, linear model, and detailed

- discussions on twisting and buckling mechanism, which include Refs. [36–41].
- [36] S. J. Stuart, A. B. Tutein, and J. A. Harrison, *J. Chem. Phys.* **112**, 6472 (2000).
- [37] F. Liu, P. Ming, and J. Li, *Phys. Rev. B* **76**, 064120 (2007).
- [38] P. Z. Hanakata, A. Carvalho, D. K. Campbell, and H. S. Park, *Phys. Rev. B* **94**, 035304 (2016).
- [39] C. Lee, X. Wei, J. W. Kysar, and J. Hone, *Science* **321**, 385 (2008).
- [40] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems, [tensorflow.org](https://www.tensorflow.org).
- [41] 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. Mach. Learn. Res.* **12**, 2825 (2011).
- [42] We perform several shuffles and find that all performances are similar (with CNN-*f16-f32-f64-h64* model). In addition, we perform ninefold cross-validation and obtain an average accuracy on the test set of 0.918 with a standard deviation of 0.002.
- [43] K. Simonyan and A. Zisserman, [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).
- [44] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Proc. IEEE* **86**, 2278 (1998).
- [45] Details of the specific GPU can be found in the SM.
- [46] N. Hu, D. Chen, D. Wang, S. Huang, I. Trase, H. M. Grover, X. Yu, J. X. J. Zhang, and Z. Chen, *Phys. Rev. Applied* **9**, 021002 (2018).