

Predicting Frictional Properties of Graphene Kirigami Using Molecular Dynamics and Neural Networks

Designs for a negative friction coefficient

Mikkel Metzsch Jensen



Thesis submitted for the degree of
Master in Computational Science: Materials Science
60 credits

Department of Physics
Faculty of mathematics and natural sciences

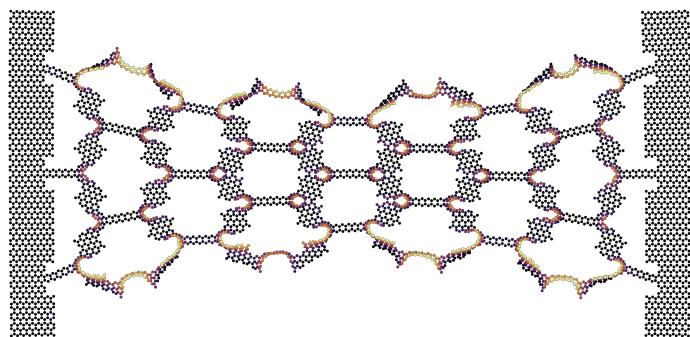
UNIVERSITY OF OSLO

Spring 2023

Predicting Frictional Properties of Graphene Kirigami Using Molecular Dynamics and Neural Networks

Designs for a negative friction coefficient

Mikkel Metzsch Jensen



© 2023 Mikkel Metzsch Jensen

Predicting Frictional Properties of Graphene Kirigami Using Molecular Dynamics and Neural Networks

<http://www.duo.uio.no/>

Printed: Reprocentralen, University of Oslo

Abstract

Various theoretical models and experimental results propose different governing mechanisms for friction at the nanoscale. We consider a graphene sheet modified with Kirigami-inspired cuts and under the influence of strain. Prior research has demonstrated that this system exhibits out-of-plane buckling, which could result in a decrease in contact area when sliding on a substrate. According to asperity theory, this decrease in contact area is expected to lead to a reduction of friction. However, to the best of our knowledge, no previous studies have investigated the friction behavior of a nanoscale Kirigami graphene sheet under strain. Here we show that specific Kirigami designs yield a non-linear dependency between kinetic friction and the strain of the sheet. Using molecular dynamics simulations, we have found a non-monotonic increase in friction with strain. We found that the friction-strain relationship does not show any clear dependency on contact area which contradicts asperity theory. Our findings suggest that the effect is associated with the out-of-plane buckling of the graphene sheet and we attribute this to a commensurability effect. By mimicking a load-strain coupling through tension, we were able to utilize this effect to demonstrate a negative friction coefficient on the order of -0.3 for loads in the range of a few nN. In addition, we have attempted to use machine learning to capture the relationship between Kirigami designs, load, and strain, with the objective of performing an accelerated search for new designs. Although this approach yielded some promising results, we conclude that further improvements to the dataset are necessary in order to develop a reliable model. We anticipate our findings to be a starting point for further investigations of the underlying mechanism for the frictional behavior of a Kirigami sheet. For instance, the commensurability hypothesis could be examined by varying the sliding angle in simulations. We propose to use an active learning strategy to extend the dataset for the use of machine learning to assist these investigations. If successful, further studies can be done on the method of inverse design. In summary, our findings suggest that the application of nanoscale Kirigami can be promising for developing novel friction-control strategies.

Acknowledgments

The task of writing a master's thesis is a demanding and extensive project which I could not have done without the support of many good people around me. First of all, I want to thank my supervisors Henrik Andersen Sveinsson and Anders Malthe-Sørensen for the assistance in this thesis work. I am especially grateful for the weekly meetings with Henrik and the inspiring discussions had as we unraveled the discoveries related to the topic of this thesis. I remember that I initially asked for an estimate of how much time he had available for supervision and the answer was something along the lines of "There are no limits really, just send me an email and we figure it out". This attitude captures the main experience I have had working with Henrik and I am profoundly grateful for the time and effort he has put into this project. I hope that he did not regret this initial statement too much, because I have certainly been taken advantage of it. I also want to thank Even Marius Nordhagen for technical support regarding the use of the computational cluster. In that context, I also want to acknowledge the Center for Computing in Science Education (CCSE) for making these resources available.

I would like to express my gratitude to all the parties involved in making it possible for me to write my thesis from Italy. I am particularly grateful for the flexibility shown by my supervisors and for the support of Anders Kvellestad, who allowed me to work remotely as a group teacher. I would also like to thank Scuola Normale Superiore for providing me with access to their library.

I realize that it is a commonly used cliché to express gratitude for the support of loved ones. However, I want to highlight the exceptional role played by my fiancé, Ida, who deserves the main credit for enabling me to maintain a healthy state of mind. She has provided me with a solid foundation for a fulfilling life that enables me to pursue secondary objectives, such as an academic career. I look forward to spending the rest of my life with you.

In this thesis, I have used the formal pronoun "we" mainly as a customary habit related to the formalities of scientific writing in a team. Nonetheless, I have realized that this usage is more fitting as I have not been working alone on this project. I have received support all the way from colleagues and friends at the University of Oslo, my family residing in Denmark, and my life partner who slept beside me every night here in Italy. They are the "good people around me" who have made this thesis possible.

Acronyms

CNN Convolutional Neural Network. 16

GAN Generative Adversarial Networks. 2

MD Molecular Dynamics. 1, 2, 3, 4, 9, 16, 20, 27, 30

ML Machine Learning. 2, 24, 26, 27, 28

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Goals	2
1.3	Contributions	3
1.4	Thesis structure	3
I	Background Theory	5
II	Simulations	7
2	Kirigami configuration exploration	9
2.1	Generating the dataset	9
2.2	Data analysis	10
2.3	Properties of interest	10
2.4	Machine learning	16
2.4.1	Architecture	16
2.4.2	Data handling	17
2.4.2.1	Input	17
2.4.2.2	Output	17
2.4.2.3	Data augmentation	17
2.4.3	Loss	17
2.4.4	Hypertuning	18
2.4.5	Model performance	22
2.5	Accelerated Search	27
2.5.1	Patteren generation search	27
2.5.2	Genetic algorithm search	31
Appendices		37

Chapter 1

Introduction

1.1 Motivation

Friction is the force that prevents the relative motion of objects in contact. In our everyday life, we recognize it as the inherent resistance to sliding motion. Some surfaces appear slippery and some appear rough, and we know intuitively that sliding down a snow-covered hill is much more exciting than its grassy counterpart. Without friction, it would not be possible to walk across a flat surface, lean against the wall without falling over or secure an object by the use of nails or screws [1, p. 5]. It is probably safe to say that the concept of friction is integrated into our everyday life to such an extent that most people take it for granted. However, the efforts to control friction date back to the early civilization (3500 B.C.) with the use of the wheel and lubricants to reduce friction in translational motion [2]. Today, friction is considered a part of the wider field *tribology* derived from the Greek word *tribos* meaning “rubbing”. It includes the science of friction, wear and lubrication [2]. The most compelling motivation to study tribology is ultimately to gain full control of friction and wear for various technical applications. Especially, the reduction of friction is of great interest since this can be utilized to improve energy efficiency in mechanical systems with moving parts. Hence, it has been reported that tribological problems have a significant potential for both economic and environmental improvements [3]:

“On global scale, these savings would amount to 1.4% of the GDP annually and 8.7% of the total energy consumption in the long term.” [4].

On the other hand, the reduction of friction is not the only sensible application for tribological studies. Controlling frictional properties, besides minimization, might be of interest in the development of a grasping robot where finetuned object handling is required. While achieving a certain “constant” friction response is readily obtained through appropriate material choices, we are yet to unlock the full capabilities to alter friction dynamically on the go. One example from nature inspiring us to think along these lines is the gecko feet. More precisely, the Tokay gecko has received a lot of attention in scientific studies aiming to unravel the underlying mechanism of its “toggable” adhesion properties. Although the gecko can produce large adhesive forces, it retains the ability to remove its feet from an attachment surface at will [5]. This makes the gecko able to achieve a high adhesion on the feet when climbing a vertical surface while lifting them for the next step remains relatively effortless. For a grasping robot, we might consider an analog frictional concept of a surface material that can change from slippery to rough on demand depending on specific tasks; slippery and smooth when interacting with people and rough and firmly gripping when moving heavy objects.

In recent years an increasing amount of interest has gone into the studies of the microscopic origins of friction, due to the increased possibilities in surface preparation and the development of nanoscale experimental methods. Nano-friction is also of great concern for the field of nano-machining where the frictional properties between the tool and the workpiece dictate machining characteristics [3]. With concurrent progress in computational capacity and development of Molecular Dynamics (MD), numerical investigations serve as an invaluable tool for getting insight into the nanoscale mechanics associated with friction. This simulation-based approach can be considered as a “numerical experiment” enabling us to create and probe a variety of high-complexity systems which are still out of reach for modern experimental methods.

In materials science such MD-based numerical studies have been used to explore the concept of so-called *metamaterials* where the material compositions are designed meticulously to enhance certain physical properties [6–11]. This is often achieved either by intertwining different material types or removing certain regions completely. In recent papers by Hanakata et al. [6, 7], numerical studies have showcased that the mechanical properties of a graphene sheet, yield stress and yield strain, can be altered through the introduction of so-called *Kirigami* inspired cuts into the sheet. Kirigami is a variation of origami where the paper is cut additionally to being folded. While these methods originate as an art form, aiming to produce various artistic objects, they have proven to be applicable in a wide range of fields such as optics, physics, biology, chemistry and engineering [12]. Various forms of stimuli enable direct 2D to 3D transformations through the folding, bending, and twisting of microstructures. While original human designs have contributed to specific scientific applications in the past, the future of this field is highly driven by the question of how to generate new designs optimized for certain physical properties. However, the complexity of such systems and the associated design space makes for seemingly intractable¹ problems ruling out analytic solutions.

Earlier design approaches such as bioinspiration, looking at gecko feet for instance, and Edisonian, based on trial and error, generally rely on prior knowledge and an experienced designer [9]. While the Edisonian approach is certainly more feasible through numerical studies than real-world experiments, the number of combinations in the design space rather quickly becomes too large for a systematic search, even when considering the computation time on modern-day hardware. However, this computational time constraint can be relaxed by the use of machine learning (ML) which has been proven successful in the establishment of a mapping from the design space to physical properties of interest. This gives rise to two new styles of design approaches: One, by utilizing the prediction from a trained network we can skip the MD simulations altogether resulting in an *accelerated search* of designs. This can be further improved by guiding the search according to the most promising candidates. For instance, as done with the *genetic algorithm* based on mutation and crossing. Another more sophisticated approach is through generative methods such as *Generative Adversarial Networks* (GAN) or diffusion models. The latter is being used in state-of-the-art AI systems such as OpenAI’s DALL-E2 [13] or Midjourney [14]. By working with a so-called *encoder-decoder* network structure, one can build a model that reverses the prediction process. This is often referred to as *inverse design*, where the model predicts a design based on physical target properties. In the papers by Hanakata et al. [6, 7] both the accelerated search and the inverse design approach was proven successful to create novel metamaterial Kirigami designs with the graphene sheet.

Hanakata et al. attribute the variation in mechanical properties to the non-linear effects arising from the out-of-plane buckling of the sheet. Since it is generally accepted that the surface roughness is of great importance for frictional properties it can be hypothesized that Kirigami-induced out-of-plane buckling can also be exploited for the design of frictional metamaterials. For certain designs, we might hope to find a relationship between the stretching of the sheet and frictional properties. If significant, this could give rise to an adjustable friction beyond the point of manufacturing. For instance, the grasping robot might apply such a material as artificial skin for which stretching or relaxing of the surface could result in a changeable friction strength.

In addition, the Kirigami graphene properties can be explored through a potential coupling between the strain and the normal load, through a nanomachine design, with the aim of altering the friction coefficient. This invites the idea of non-linear friction coefficients which might in principle also take on negative values. This would constitute a rarely found property which is mainly observed for the unloading phase of adhesive surfaces [15] or in the loading phase of particular heterojunction materials [16, 17].

To the best of our knowledge, Kirigami has not yet been implemented to alter the frictional properties of a nanoscale system. However, in a recent paper by Liefferink et al. [18] it is reported that macroscale Kirigami can be used to dynamically control the macroscale roughness of a surface through stretching. They reported that the roughness change led to a changeable frictional coefficient by more than one order of magnitude. This supports the idea that Kirigami designs can be used to alter friction, but we believe that taking this concept to the nanoscale would involve a different set of governing mechanisms and thus contribute to new insight in this field.

1.2 Goals

In this thesis, we investigate the prospects of altering the frictional properties of a graphene sheet through the application of Kirigami-inspired cuts and stretching of the sheet. With the use of molecular dynamics (MD)

¹In computer science we define an *intractable* problem as a problem with no *efficient* algorithm to solve it nor any analytical solutions. The only way to solve such problems is the *brute-force* approach, simply trying all possible combinations, which is often beyond the capabilities of computational resources.

simulations, we evaluate the frictional properties of various Kirigami designs under different physical conditions. Based on the MD results, we investigate the possibility to use machine learning for the prediction of frictional properties and subsequently using the model for an accelerated search of new designs. The main goals of the thesis can be summarized as follows.

1. Design an MD simulation procedure to evaluate the frictional properties of a Kirigami graphene sheet under specified physical conditions.
2. Develop a numerical tool to generate various Kirigami designs, both by seeking inspiration from macroscale designs and by the use of a random-walk-based algorithm.
3. Investigate the frictional behavior under varying strain and load for different Kirigami designs.
4. Develop and train a machine learning model to predict the MD simulation results and perform an accelerated search of new designs with the goal of optimizing certain frictional properties.

1.3 Contributions

By working towards the goals outlined above (Sec. 1.2), I have discovered a non-linear relationship between the kinetic friction and the strain for certain Kirigami patterns. This phenomenon was found to be associated with the out-of-plane buckling of the Kirigami sheet but with no clear relationship to the contact area or the tension in the sheet. I found that this method does not provide any mechanism for a reduction in friction, in comparison to a non-cut sheet. However, the straining of certain Kirigami sheets allows for a non-monotonic increase in friction. The relationship to normal load was proven negligible in this context and I have demonstrated that a coupled system of load and strain (through sheet tension) can exhibit a negative friction coefficient in certain load ranges. Moreover, I have created a dataset of roughly 10,000 data points for assessing the employment of machine learning and accelerated search of Kirigami designs. I have found, that this approach might be useful, but that it requires an extended dataset in order to produce reliable results for a search of new designs.

During my investigations, I have built three numerical tools, in addition to the usual scripts for data analysis, which are available on Github [19]. The tools are summarized in the following.

- I have written a LAMMPS-based [20] tool for simulating and measuring the frictional properties of a graphene sheet sliding on a substrate. The code is generally made flexible with regard to the choice of sheet configuration, system size, simulation parameters and MD potentials, which makes it applicable for further studies on this topic. I have also built an automated procedure to carry out multiple simulations under varying parameters by submitting jobs to a computational cluster via an ssh connection. This was done by adding minor additions to the Python package developed by E. M. Nordhagen [21].
- I have generated a Python-based tool for generating Kirigami patterns and exporting these in a compatible format with the simulation software created. The generation of molecular structures is done with the use of ASE [22]. Our software includes two classes of patterns inspired by macroscale designs and a random walk algorithm which allows for a variety of different designs through user-defined biases and constraints. Given our system size of choice, the first two pattern generators are capable of generating on the order of 10^8 unique designs while the random walk generator allows for significantly more.
- I have built a machine-learning tool based on Pytorch [23] which includes setting up the data loaders, a convolutional network architecture, a loss function, and general algorithms for training and validating the results. Additionally, I have written several scripts for performing grid searches and analyzing the model predictions in the context of the frictional properties of graphene.

All numerical implementations have been originally developed for this thesis except for the libraries mentioned above along with common Python libraries such as Numpy and Matplotlib.

1.4 Thesis structure

The thesis is divided into two parts. In Part I we introduce the relevant theoretical background, and in Part II we present the numerical implementations and the results of this thesis.

Part I contains a description of the theoretical background related to Friction (??), Molecular Dynamics (??) and Machine Learning (??). In ?? we formulate our research questions in the light of the friction theory.

In Part II, we begin by presenting the system in ?? which includes a definition of the main parts of the system and the numerical procedures related to the MD simulation. Here we also present the generation of Kirigami designs. In ?? we carry out a pilot study where we evaluate the simulation results for various physical conditions and compare a non-cut sheet to two different Kirigami designs. In Chapter 2, we further explore the Kirigami patterns through the creation of a dataset and the employment of machine learning and an accelerated search for new designs. In ??, we use the results from the pilot study to demonstrate the possibility to achieve a negative friction coefficient for a system with coupled load and strain. Finally, in ??, we summarize our results and provide an outlook for further studies. Additional figures are shown in ??, ?? and ??.

Part I

Background Theory

Part II

Simulations

Chapter 2

Kirigami configuration exploration

Building upon the discoveries of the Pilot Study ??, we will further explore the impact of Kirigami designs on strain-dependent friction. Our focus is primarily to optimize the friction force and friction coefficient toward their maximum or minimum values. To achieve this goal, we will utilize MD simulations to generate an extended dataset that encompasses a wider range of Kirigami designs. This is motivated by the aim of gaining a broader understanding of the friction-strain relationship. We will then leverage this dataset to explore the potential of employing machine learning for the prediction of friction behavior based on Kirigami design, strain, and load. Finally, we utilize the developed machine learning model for an accelerated search for new Kirigami designs.

2.1 Generating the dataset

We create a dataset that contains an extended series of Kirigami design configurations based on the pattern generation methods developed in ?? for which we will vary the strain and load for each configuration. For each configuration, we sample 15 strain values between 0 and the rupture strain using a pseudo-uniform distribution, meaning that we divide the given interval into equal segments and pick a value from each segment by a uniform distribution. This is due to numerical limitations in LAMMPS², but we find that this gives evenly spaced values which also carry some randomness. Since the normal load did not prove to be dominant in the friction description we only sample 3 normal load values per configuration, uniformly sampled in the range [0.1, 10] nN. In total, this gives $3 \times 15 = 45$ data points for each configuration. For the remaining parameters, we use the default values shown in ???. We are mainly concerned with the mean friction and whether the sheet ruptures during the simulation. However, we also include the maximum friction, the relative contact, the rupture strain (from the rupture test) and the porosity (void fraction) in the dataset. We generate 68 configurations of the Tetrahedron pattern type, 45 of the Honeycomb type and 100 of the Random walk type. For the Tetrahedron and Honeycomb patterns, we choose a random reference position which results in translational variances of the patterns. A summary of the dataset is given in Table 2.1 while all configurations are shown visually in ???. The Tetrahedron and Honeycomb parameters are chosen to provide additional variations of the configurations evaluated in ?? which exhibited interesting properties. The Random walk parameters are chosen to introduce as much variety as possible in order to contribute to a wide distribution of configurations in the dataset. Notice that not all submitted data points “make it” to the final dataset, which is due to a small bug in the data generation procedure³.

²In LAMMPS, we sample the various strain values by storing restart files during the straining of the sheet. The restart values are stored at specific timesteps governed by a LAMMPS variable. Such variables allow for a vector of uniform randomly chosen values, but unfortunately, we are not able to sort the vector for ascending values. This will lead to the script waiting to store the restart file according to the next timestep value in the unsorted vector. As soon as the next timestep value is less than the current timestep the program will stop producing restart files and thus skip most of them. However, by first defining a series of intervals we can draw a uniform number for each interval without getting into trouble.

³The issue arises from the fact that the rupture point in the rupture test does not always match the rupture point in the following simulations. After performing the rupture test the simulation is restarted with a new substrate size, corresponding to the measured rupture strain limit, but also with a new random velocity and thermostat initialization. The sheet is then strained and checkpoints of the simulation state, restart files, are stored for each of the targeted strain samples. However, if the rupture point arrives earlier than suggested by the rupture test, due to randomness from the initialization, some of the planned strain samples do not get a corresponding restart file. Thus, these data points are not included in the dataset even though they ideally should have been noted as a rupture event. This could have been mitigated by a rewrite of the code, but it was first discovered after the dataset had been

Table 2.1: Summary of the number of generated data points in the dataset. Due to slight deviations in the rupture strain and the specific numerical procedure not all submitted simulations are included in the final dataset. Notice that the Tetrahedon (7, 5, 2) and Honeycomb (2, 2, 1, 5) from the pilot study are rerun as a part of the Tetrahedon and the Honeycomb datasets separately. However, the reference point for the patterns is randomized and thus these configurations are not fully identical. This is the reason for the ambiguousness in the total sum.

Type	Configurations	Submitted data points	Final data points	Ruptures
Pilot study	3	270	261	25 (9.58 %)
Tetrahedon	68	3060	3015	391 (12.97 %)
Honeycomb	45	2025	1983	80 (4.03 %)
Random walk	100	4500	4401	622 (14.13 %)
Total	214 (216)	9855	9660	1118 (11.57 %)

2.2 Data analysis

In order to gain insight into the correlations in the data we calculate the correlation coefficients between all variable combinations. More specifically, we calculate the Pearson product-moment correlation coefficient which is defined, between data set X and Y , as

$$\text{corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\langle (X - \mu_X)(Y - \mu_Y) \rangle}{\sigma_X \sigma_Y} \in [-1, 1], \quad (2.1)$$

where $\text{Cov}(X, Y)$ is the covariance, μ the mean value and σ the standard deviation. The correlation coefficients range from a perfect negative correlation (-1) through no correlation (0) to a perfect positive correlation (1). The correlation coefficients are shown in Fig. 2.1. We especially notice that the mean friction force $\langle F_{\parallel} \rangle$ has a significant positive correlation with strain (0.77) and porosity (0.60). However, the relative strain, scaled by the rupture strain, has a weaker correlation of only 0.25. This indicates that the correlation might be associated with the flexibility of the configurations since these can be taken to higher absolute values of strain. This is further supported by the fact that the mean friction and the rupture strain are also strongly positively correlated (0.78). We also observe that the contact is negatively correlated with the mean friction (-0.67) and the strain value (-0.74). This is generally consistent with the trend observed in the pilot study in ?? where the increasing strain was associated with a decreasing contact and mainly increasing mean friction. However, we must note that the correlation coefficient is a measure of the quality of a forced linear fit on the data. Since we have observed a non-linear trend between friction and strain (??) we should not expect any near 100% correlations. Additionally, we also notice that all correlations to normal load are rather low, which aligns well with the findings in the pilot study.

Fig. 2.2 shows a visualization of the data (excluding the pilot study configurations) for a subset of variable pairs on the axes. This allows us to visually identify some of the correlations and gain a qualitative understanding of the variations in different planes of the feature space which we eventually will base our machine learning model on.

2.3 Properties of interest

In the pilot study (??) we found promising results for the idea of achieving a negative friction coefficient under the assumption of a system with coupled normal load F_N and strain ε . Hence, we will consider this as a main property of interest for our further exploration. We assume that the friction dependence on load is negligible $F(F_N, \varepsilon) \sim F(\varepsilon)$ in comparison to that on strain, and propose a coupling $\varepsilon = RF_N$ with linear coupling ratio R . From these assumptions, we can in practice substitute load for strain in the expression for the friction coefficient of our coupled system $\mu \propto \Delta F_f(\varepsilon)/\Delta\varepsilon$ as shown in ???. This justifies the search for a negative slope on the friction-strain curve since this can be related to a negative friction coefficient in our proposed coupled system.

created. Despite the issue, the dataset still contains a notable 11.57% of rupture events, which is deemed sufficient for the machine learning model to learn to identify ruptures. Therefore we conclude that this issue is not critical for machine learning training.

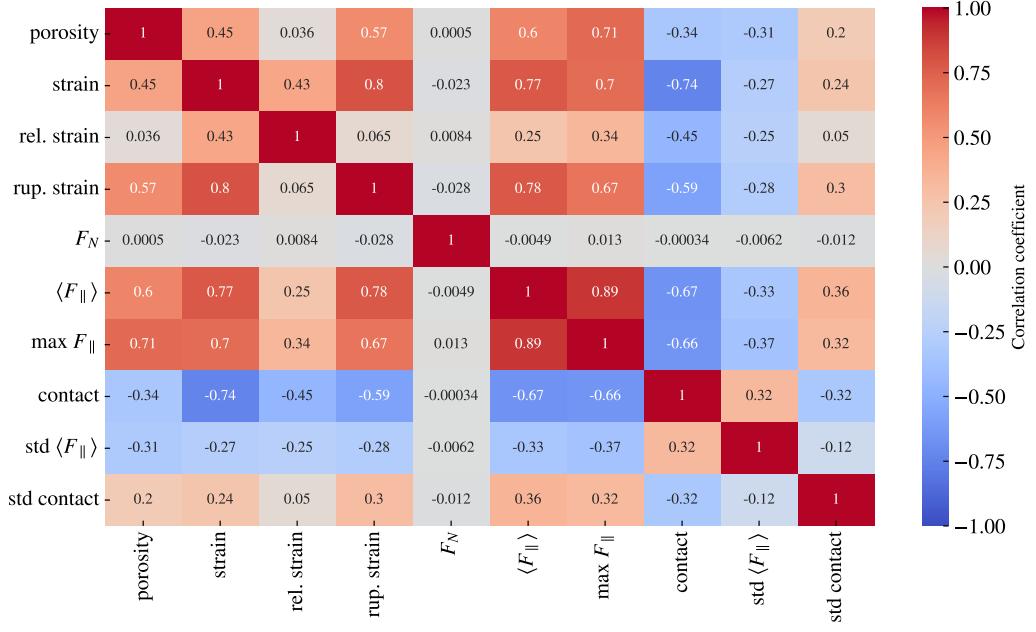


Figure 2.1: Pearson product-moment correlation coefficients Eq. (2.1) for the full datset (see Table 2.1). Here the relative strain refers to the strain relative to the rupture strain.

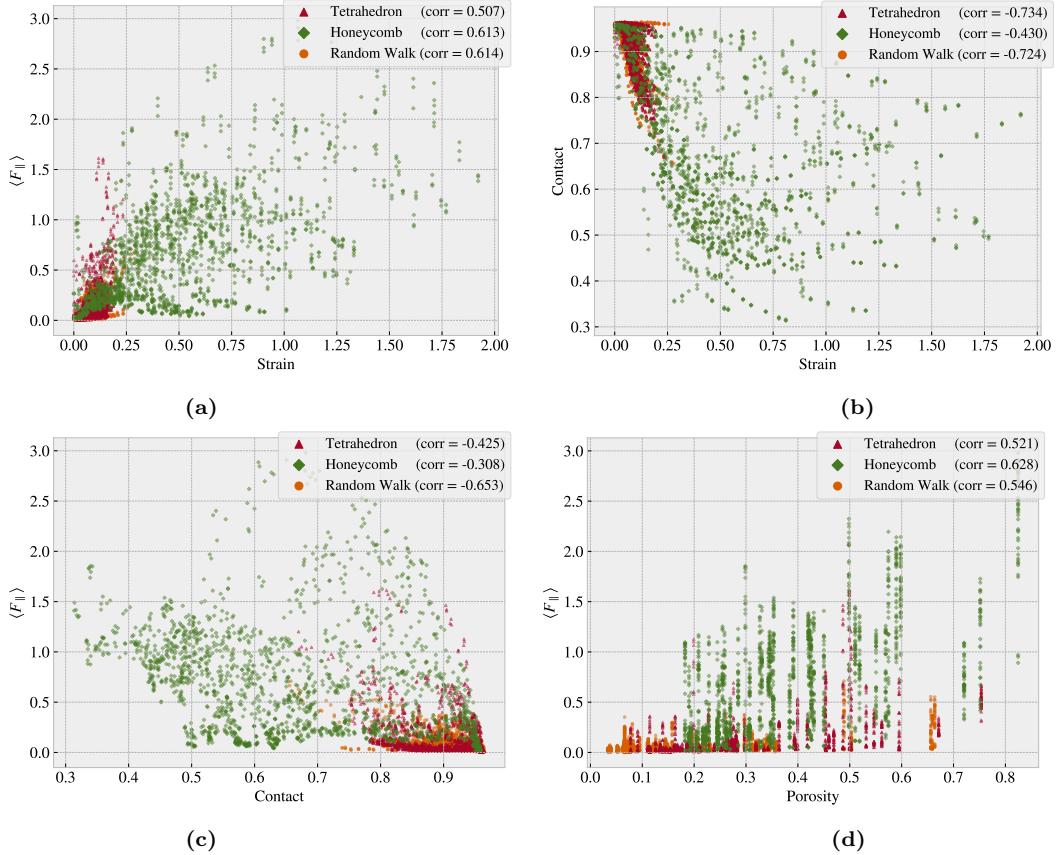


Figure 2.2: Scatter plot of various dataset feature pairs for the Tetrahedron, Honeycomb and Random Walk subsets respectively. The legends include the correlation coefficient in parentheses. (a) Mean friction vs. strain. (b) Relative contact vs. strain. (c) Mean friction vs. relative contact. (d) Mean friction vs. porosity.

The remaining question is then how to evaluate the strength of this property. By definition, the minimum (most negative) slope value would give the lowest friction coefficient. However, two data points with a small $\Delta\epsilon$, corresponding to a small denominator in ??, would potentially lead to a huge negative slope value without any significant decrease in friction. Hence, we choose to consider the decrease in friction with increasing strain as a better metric. Numerically we compute this by locating the local maxima on the friction-strain curve and then evaluating the difference to the succeeding local minima. The biggest difference corresponds to the *max drop* property which serves as our indicator for a negative friction coefficient. In this evaluation, we do not guarantee a monotonic decrease of friction in the strain range corresponding to the max drop, but when searching among multiple configurations this is considered a decent strategy to highlight configurations of interest worthy of further investigation. In addition to the max drop property, we also consider the minimum, $\min F_{\text{fric}}$, the maximum, $\max F_{\text{fric}}$ and the maximum difference, $\max \Delta F_{\text{fric}} = \max F_{\text{fric}} - \min F_{\text{fric}}$ for the friction-strain curve. The extrema of these four properties for each of the categories: Tetrahedron, Honeycomb, Random walk and pilot study, are summarized in Table 2.2. The corresponding friction-strain profiles and configurations are shown in Fig. 2.3a to 2.3d (excluding the friction-strain profiles already shown in the pilot study ??). The friction-strain profiles for the full dataset are shown in ??.

Table 2.2: Evaluation of the properties of interest for the dataset. Each table shows the top scores for each of the four properties within each of the separate data categories: Tetrahedron, Honeycomb, Random walk and pilot study (the three configurations used in the pilot study). The tables denote the names of the top candidate configurations, the relevant strain values for the property and the property values themselves. For the Tetrahedron and Honeycomb category, we compare the top candidate scores to the scores of the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) pattern used in the pilot study in the right-most column.

Tetrahedron	Configuration	Strain	Value [nN]	Tetrahedon (7, 5, 1) [nN]
$\min F_{\text{fric}}$	(3, 9, 4)	0.0296	0.0067	0.0262
$\max F_{\text{fric}}$	(5, 3, 1)	0.1391	1.5875	0.8891
$\max \Delta F_{\text{fric}}$	(5, 3, 1)	[0.0239, 0.1391]	1.5529	0.8603
max drop	(5, 3, 1)	[0.1391, 0.1999]	0.8841	0.5098

Honeycomb	Configuration	Strain	Value [nN]	Hon. (2, 2, 1, 5) [nN]
$\min F_{\text{fric}}$	(2, 5, 1, 1)	0.0267	0.0177	0.0623
$\max F_{\text{fric}}$	(2, 1, 1, 1)	1.0654	2.8903	1.5948
$\max \Delta F_{\text{fric}}$	(2, 1, 5, 3)	[0.0856, 1.4760]	2.0234	1.5325
max drop	(2, 3, 3, 3)	[0.5410, 1.0100]	1.2785	0.9674

Random walk	Configuration	Strain	Value [nN]
$\min F_{\text{fric}}$	12	0.0562	0.0024
$\max F_{\text{fric}}$	96	0.2375	0.5758
$\max \Delta F_{\text{fric}}$	96	[0.0364, 0.2375]	0.5448
max drop	01	[0.0592, 0.1127]	0.1818

Pilot study	Configuration	Strain	Value [nN]
$\min F_{\text{fric}}$	No cut	0.2552	0.0012
$\max F_{\text{fric}}$	Hon. (2, 2, 1, 5)	0.7279	1.5948
$\max \Delta F_{\text{fric}}$	Hon. (2, 2, 1, 5)	0.7279	1.5325
max drop	Hon. (2, 2, 1, 5)	[0.7279, 1.0463]	0.9674

From the property comparison in Table 2.2, we find that both the Tetrahedron and Honeycomb subsets contain improved candidates for each of the property scores in comparison to the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) examined in the pilot study. Overall, the Honeycomb pattern type is resulting in the highest scores for the maximum properties while the minimum friction is still achieved by the non-cut sheet. This latter observation confirms the findings of the pilot study since our dataset does not provide any indication that friction can be reduced for a Kirigami sheet under strain. However, the improvement in the remaining properties indicates that the dataset contains valuable information that can provide a direction for further optimization of the maximum properties. Considering the Random walk we find that the max property scores

are generally lower than those of the Tetrahedron and Honeycomb patterns. However, since these are found to be on a comparable order of magnitude we argue that these contribute relevant information for the frictional dependency to Kirigami configurations. The Random walk patterns exhibit greater diversity compared to the other patterns and therefore provide some immediate insights into which structures can be associated with each of the properties of interest. For the min F_{fric} top candidates (Fig. 2.3a) we find that the Random walk candidate has a rather low cut density (low porosity) and vertical cuts. Since these cuts run parallel to the stretching direction one can hypothesize that this minimizes the induced buckling effect which agrees with the relatively flat contact-strain curve. For the minimum candidate of the Tetrahedron pattern, we also observe a low decrease in contact area, and in both these cases this corresponds with a seemingly flat friction-strain curve as well. When considering the remaining friction-strain curves throughout Fig. 2.3a to 2.3d we find that a rising friction-strain curve is always seen together with a declining contact-strain curve. This supports the general observation of a correlation between the strain-induced friction effects and the contact area. When looking at the 96th Random walk pattern, which is the top candidate for both the max F_{fric} (Fig. 2.3b) and max Δf_{fric} (Fig. 2.3c) properties, we find a rather porous configuration with mainly horizontal-orientated cuts. This has some structural reminiscence with the general shape of the Honeycomb pattern. Finally, for the Random walk max drop candidate, Random walk pattern 01, we do see a small drop in friction. Although, this is not as significant as seen for the Tetrahedron and Honeycomb candidates. We notice that the configuration contains some slanted cuts which might be reminiscent of parts of the general Tetrahedron pattern.

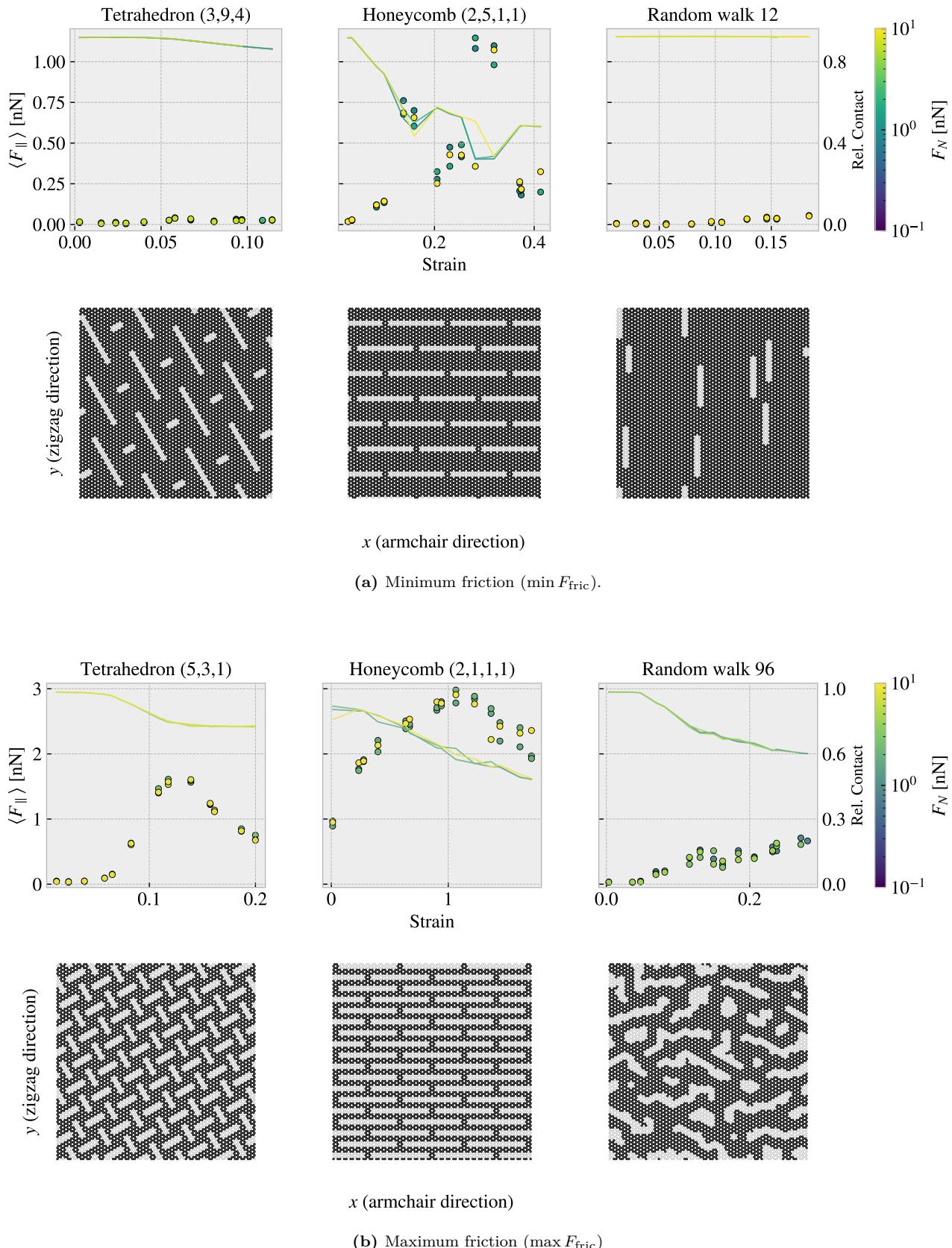


Figure 2.3: (The figure continues on the next page)

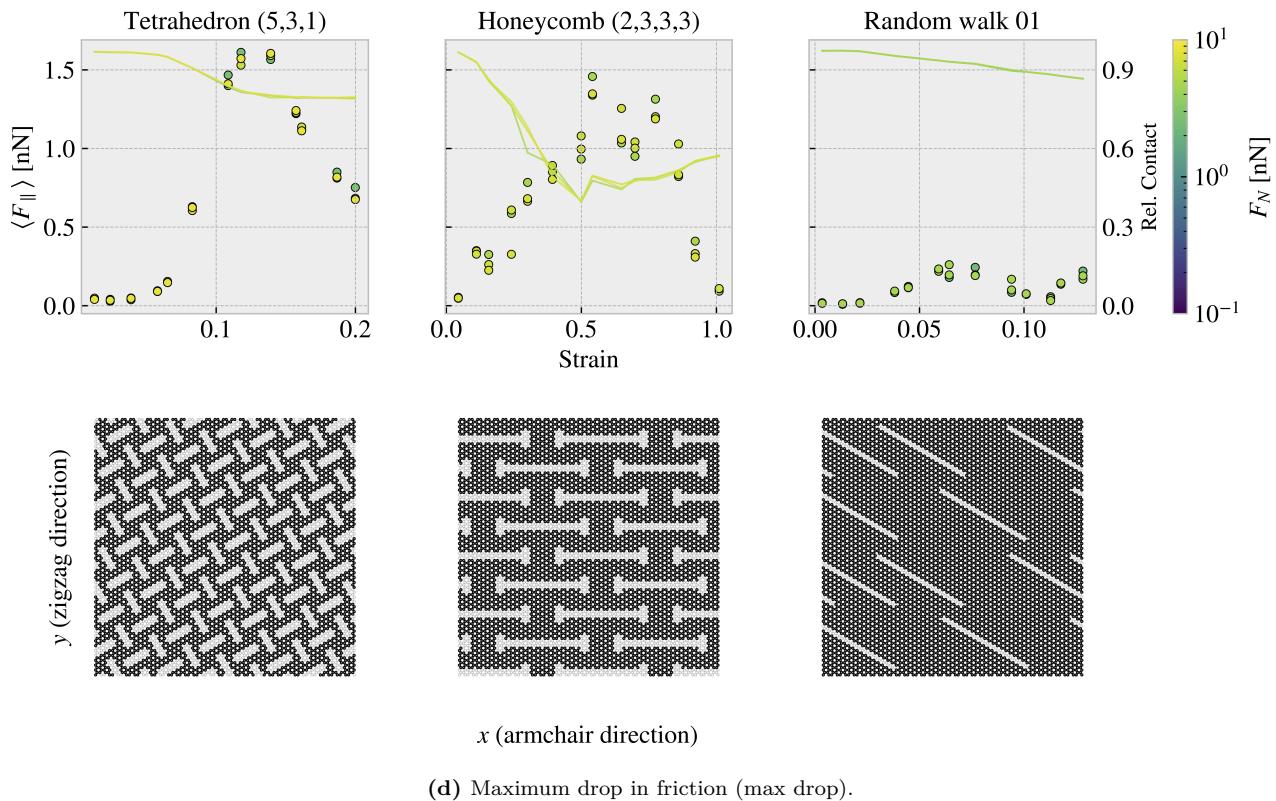
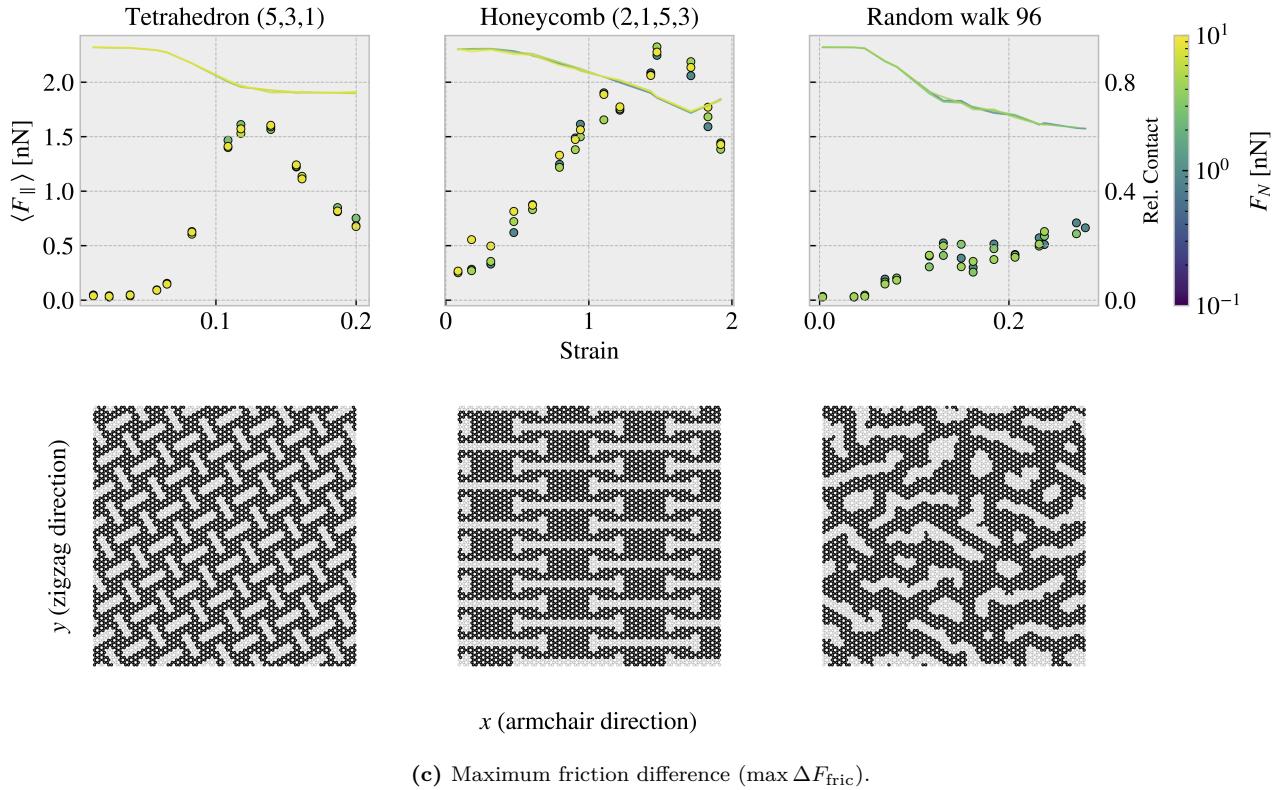


Figure 2.3: Illustration of the top candidates within each of the four properties of interest. The upper row shows the friction-strain curve with dotted points (friction value on the left y-axis) and the contact-strain curve with a solid line (relative contact value on the right y-axis). The bottom row shows the corresponding Kirigami patterns. (a) Minimum friction ($\min F_{\text{fric}}$). (b) Maximum friction ($\max F_{\text{fric}}$). (c) Maximum friction difference ($\max \Delta F_{\text{fric}}$). (d) Maximum drop in friction (max drop).

2.4 Machine learning

Given the MD-based dataset presented in the previous section, containing Honeycomb, Tetrahedron and Random walk geometries, we investigate the possibilities of training a machine learning model to predict the friction behavior from a given Kirigami configuration, strain and load.

2.4.1 Architecture

Due to the spatial dependencies in the Kirigami configurations, we use a convolutional neural network (CNN). Similar studies which predict mechanical properties for graphene sheets have used a VGGNet style of network, Hanakata et al. [6, 7] and Wan et al. [8], which we adopt for this study as well. The VGGNet-16 architecture illustrated in Fig. 2.4 shows the key features that we will include:

- The image is processed through a series of 3×3 convolutional filters (the smallest size capable of capturing spatial dependencies) using a stride of 1 with an increasing number of channels throughout the network. We use zero padding to conserve the spatial size during a convolution. Each convolutional layer is followed by a ReLU activation function.
- The spatial dimensions are reduced by a max pooling layer, filter size 2×2 and a stride of 2, which halves the spatial resolution each time.
- The latter part of the network consists of a fully connected part using the ReLU activation as well. The transition from the convolutional to the fully connected part is achieved by applying a filter with the same dimensions as the last convolutional feature map. This essentially performs a linear mapping from the spatial output to the fully connected layer where the number of channels corresponds to the nodes in the first fully connected layer.

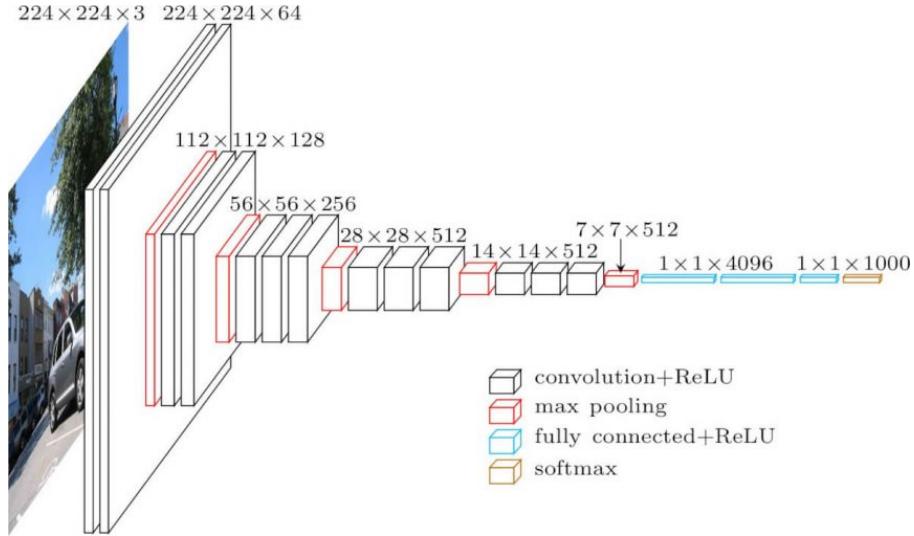


Figure 2.4: Illustration of the convolutional network architecture for the VGGNet-16 proposed by K. Simonyan and A. Zisserman [24]. Reproduced from [25].

We deviate from the VGGNet-16 architecture by including batch normalization and restricting ourselves to setting up the convolutional part of the network in terms of the convolutional block (Convolution → Batch normalization → ReLU → Max pooling). Similarly, we define a fully connected block by two elements (Fully connected → ReLU) which match the VGGNet model. Hanakata et al. and Wan et al. used a similar architecture with the parameters

$$\begin{array}{ll} \text{Hanakata et al. [6]} & C16 C32 C64 D64, \\ \text{Wan et al. [8]} & C16 C32 D32 D16, \end{array}$$

where C denotes a convolutional block with the number denoting the number of channels, and D a fully connected (dense) block with the number denoting the number of nodes. For the purpose of determining a suiting complexity for the architecture, we adopt the approach by Wan et al. [8] who used a “staircase” pattern for combining the convolutional and fully connected blocks. By defining a starting number of channels S for the first convolutional layer and a network depth D we fill the first half of the network layers with convolutional blocks, doubling in channel number for each layer, and the latter half with fully connected blocks halving the number of nodes in a reverse pattern. For instance, the architecture $S4D8$ will take the form

$$\text{Input} \rightarrow \underbrace{C4 \ C8 \ C16 \ C32 \ D32 \ D16 \ D8 \ D4}_{\substack{S=4 \\ D=8}} \rightarrow \text{Output}. \quad (2.2)$$

This provides a simple description where S and D can be varied systematically for a grid search over architecture complexity.

2.4.2 Data handling

2.4.2.1 Input

We use three variables as input: Kirigami configuration, strain of the sheet and applied normal load. The Kirigami configuration is given as a two-dimensional binary matrix while the strain and load are both scalar values. This gives rise to two different options for the data structure:

1. Expand the scalar values (strain and load) into 2D matrices of the same size as the Kirigami configuration matrix by copying the scalar value to all matrix coordinates. This can then be merged into an image of three channels used as a single input.
2. Pass only the Kirigami configuration through the convolutional part of the network and introduce the remaining scalar values directly into the fully connected part of the network halfway in.

Both options utilize the same data, but the latter option is more directed toward independent processing of the data while the first makes for an intertwined use of the configuration, strain and load input. We implemented both options but found immediately that option 1 was producing the most promising results during short training tests, and thus we settled for this data structure.

2.4.2.2 Output

For the output, we are mainly concerned with mean friction and the rupture detection. In combination, these values will make the model able to produce a friction-strain curve with an estimated stopping point as well. However, in order to retain the option to explore other relations in the data we include the maximum friction, relative contact, porosity and rupture strain in the output as well. Notice that we weigh the importance of these output variables differently in the loss as described in Sec. 2.4.3.

2.4.2.3 Data augmentation

In order to increase the utility of the available data one can introduce data augmentation. For most classification tasks this usually includes distortions such as color shifts, zoom, reflect etc. However, such distortions are only valid since the classification network should still classify a cat as a cat even though it is suddenly a bit brighter or flipped upside down. For our problem, we can only use augmentation that matches a physical symmetry. Such a symmetry exists for reflection across the y-axis, and thus we perform this transformation with a 50% chance for both training and validation data. We cannot use a reflection across the x-axis as the sheet is sliding in a positive y-direction. Such a transformation would correspond to a change in the sliding direction which we cannot expect to be fully symmetric.

2.4.3 Loss

The output contains two different types of variables: scalar values and a binary value (rupture). For the scalar values we use the mean squared error ?? and for the binary output we use binary cross entropy ???. We calculate

the total loss as a weighted sum of the loss associated with each output

$$L_{tot} = \sum_o W_o \cdot L_o.$$

We choose the weights W_o to be 1/2 for the mean friction and 1/10 for the remaining 5 output variables, thus sharing the loss evenly for the remaining 50% of the weight. During the introductory phase of the model implementation, we tried different settings for these weights, but we found that the results varied little. Hence, we concluded that this was of minor importance and we settled on the values defined above.

2.4.4 Hypertuning

For the hypertuning we focus on architecture complexity, learning rate, momentum and weight decay. We use the ADAM optimizer with the initial default values of $\beta_1 = 0.9$, $\beta_2 = 0.999$ and zero weight decay, for which we will vary momentum β_1 and weight decay in the hypertuning. We use a batch size of 32 and train the model for a maximum of 1000 epochs while storing the best model based on the validation scores. Since the learning rate is considered to be one of the most important hyperparameters we will determine a suitable choice for the learning rate using the learning rate range test for each of the two grid searches:

1. Architecture complexity grid search of S vs. D with individually chosen learning rates for each complexity combination.
2. Momentum vs. weight decay grid search with learning range chosen with regard to each momentum setting.

We consider first the architecture complexities in the range $S \times D = \{2, 4, 8, 16, 32, 64, 128, 256\} \times \{4, 6, 8, 10, 12, 14\}$. For each architecture complexity, we perform an initial learning rate range test and determine the suitable choice for the learning rate as the point for which the validation loss decreases most rapidly. The learning rate is increased exponentially within the range 10^{-7} to 10 with increments for each training batch iteration. This is done for a single epoch where a batch size of 32 yields a total of 242 possible increments. This corresponds to an exponent increment of approximately 1/30 giving a relative increase $10^{1/30} \sim 108\%$ per batch iteration. The learning rate range test is presented in Fig. 2.5 for various model complexities. We notice that the suggested learning rate decreases with an increasing number of model parameters. This decrease is further independent of the specific relationship between S and D .

With the use of the suggested learning rates from Fig. 2.5 we perform a grid search over the corresponding S and D parameters. We evaluate both the validation loss and the mean friction R^2 score for the validation data which is shown in Fig. 2.6 together with the best epoch and the number of model parameters. Additionally, we evaluate the mean friction R^2 score for a selected set of configurations. This set consists of the top 10 configurations with respect to the max drop property for the Tetrahedron and Honeycomb patterns respectively. This is done as a way of evaluating the performance on the non-linear friction-strain curves which we find to be the more difficult trend to capture. The selected evaluation is shown in Fig. 2.7. Note that these configurations are already a part of the full dataset and thus the data points related to these configurations are most likely present in both the training and the validation data set. Hence, the performance must be considered in conjunction with the actual validation performance in Fig. 2.6.

From the validation scores in Fig. 2.6, looking at both the loss and the R^2 scores, we find that models S(8-32)D(8-12) generally give the best performance. When considering at the best epoch, we find that models of low depth result in a later best epoch in the range $\sim [800, 1000]$, in comparison to models of high depth yielding the best epoch in the range $\sim [300, 600]$. This indicates a transition from underfitting to overfitting of the model since the best validation scores are found earlier in the training process. However, since our training stores the best model during training, we do not have to worry too much about overfitting. Nonetheless, we can take this transition as a sign that our search is conducted in an appropriate complexity range. When consulting the evaluation on the selected sets in Fig. 2.7 we find significantly lower R^2 scores, especially for the Tetrahedron pattern. This observation indicates that the prediction of these configurations is more difficult, especially when considering that some of these data points are already included in the training data. While the peak R^2 value for the validation score in Fig. 2.6 was found for the model S16D10 model (98.23 %) the selected set test shows a slight preference for more complexity in the model. In the Tetrahedron selected set grid search, we find the best model to be S32D12, with an R^2 score of $\sim 87\%$. This model choice is more or less compatible with the overall performance since it is among the top candidates for the R^2 score and loss in Fig. 2.6 and the R^2 score for the

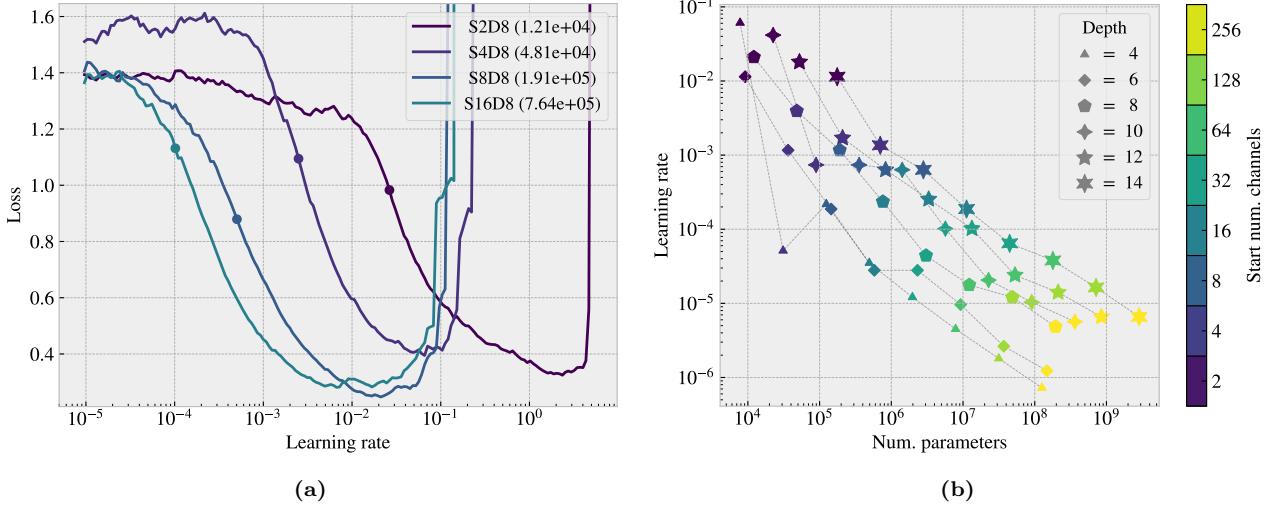


Figure 2.5: Learning rate range test for various model complexities. We increase the learning rate exponentially from 10^{-7} to 10 during one epoch corresponding to an exponent increment of roughly 1/30 per batch iteration. (a) A few examples of the validation loss as a function of the learning rate. The exemplary architectures are S[2, 16]D8 with the corresponding number of model parameters shown in parentheses in the legend. The dots indicate the suggested learning rate at the steepest decline of the validation loss. (b) The full results showing the suggested learning rates depending on the number of model parameters with color coding differentiating the number of start channels S and marker types differentiating different model depths D .

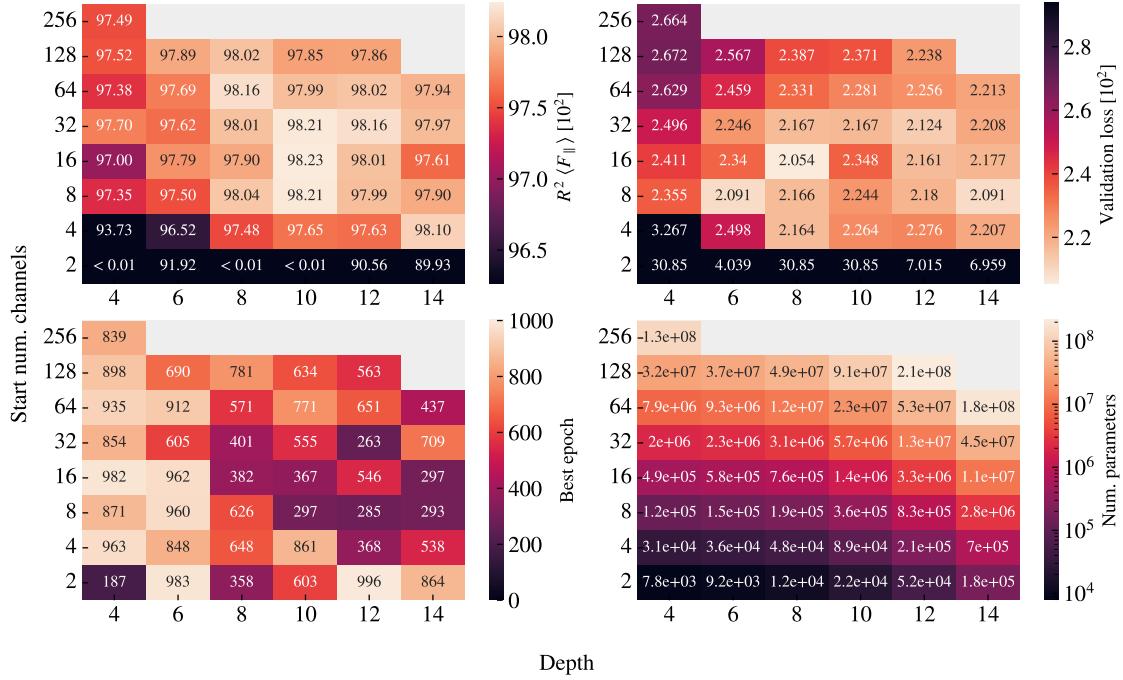


Figure 2.6: Architecture complexity grid search using a staircase-like VGGNet structure. The x-axis denotes the number of layers in the network (Depth) and the y-axis is the starting number of channels. See Eq. (2.2) for an example of the staircase architecture. For each architecture complexity, we evaluate the friction mean R^2 validation score (top left), the validation loss (top right), the best epoch stored based on validation scores (bottom left) and the number of model parameters (bottom right).

selected Honeycomb set in Fig. 2.7 as well. Hence, we settle for this architecture. We note that the theoretical receptive field for the last convolutional layer (layer 6) is 13×13 according to ??, and thus each node in the

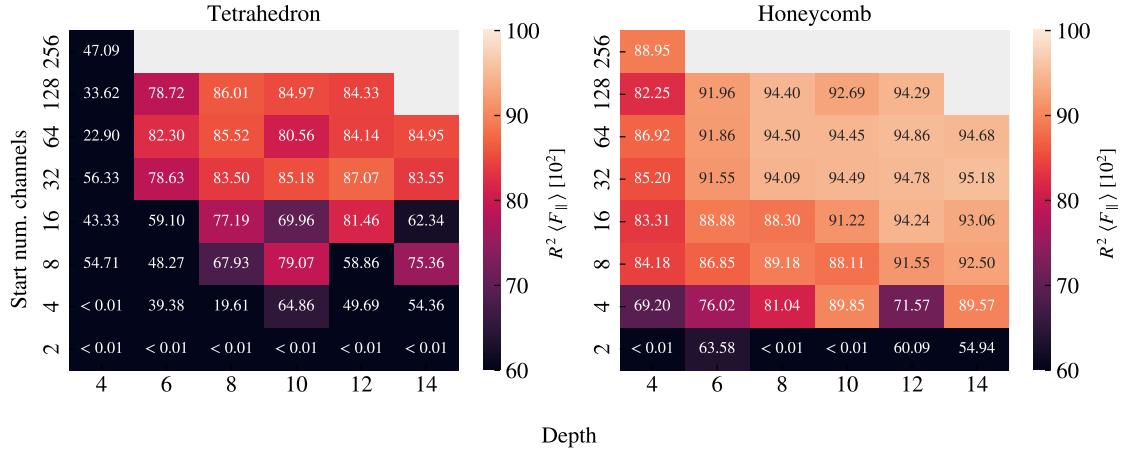


Figure 2.7: Architecture grid search similar to that of Fig. 2.6 on a selected set of configurations. The selected set consists of the top 10 candidates for the max drop property for the Tetrahedron and Honeycomb categories in the dataset respectively.

first fully connected layer does not connect to the entirety of the 62×106 input image. Hence, some spatial dependence will have to be encoded in the fully connected part as well, and we note that an enlargement of the receptive field might serve as an interesting suggestion for an improvement of the model in further studies. One possible method is to increase the stride or use dilated convolutions.

Next, we consider momentum m and weight decay λ in the range $m \in [0.85, 0.99]$ and $\lambda \in [0, 10^{-2}]$. For each choice of momentum, we perform a learning rate range test. We propose two learning rate schemes: A constant learning rate scheme as used until this point and a one-cycle policy cyclic scheme. In the cyclic scheme we set maximum bound for which the learning rate starts from a factor 1/20 of the maximum bound, increases toward the maximum bound during the first 30% of training and decreases toward a factor 10^{-4} of the maximum bound for the remaining 70% of training. The increase and decrease are done by a cosine function. The suggested learning rate for the constant learning rate scheme is once again determined by the steepest slope on the learning range test loss curve while the maximum bound used for the cyclic scheme is determined as the point just before divergence. We find that the minimum point on the loss curve is a suitable choice that approaches the diverging point without getting too close and causing instabilities in the training. The learning rate range test for momentum is shown in Fig. 2.8. We observe generally that a higher momentum corresponds to a higher suggested learning rate for both schemes. Using these results we perform a grid search of momentum and weight decay. We examine again the validation loss and validation mean friction R^2 score in addition to the friction mean R^2 score for the selected set of Tetrahedron and Honeycomb patterns. This is shown for the constant learning rate scheme in Fig. 2.9 and for the cyclic scheme in Fig. 2.10.

The original validation scores, before varying momentum and weight decay, were a validation loss of 0.02124 and a mean friction R^2 score of 0.9816. By varying momentum and weight decay, we find that these scores can be improved slightly for the constant learning rate scheme (loss: 0.02038, R^2 : 0.9823) and even more for the cyclic scheme (loss: 0.0176, R^2 : 0.9831). Note that the loss and R^2 scores here do not correspond to the same hyperparameter choices. The comparison among best scores is summarized in Table 2.3. In general, the constant scheme shows rather stable results for all momentum settings $m \in [0.85, 0.99]$ in combination with a low weight decay $\lambda \leq 10^{-4}$. For the cyclic scheme the performance peaks toward a low momentum $m \leq 0.93$ and a low weight decay $\lambda \leq 10^{-4}$. Looking at the summary in Table 2.3, we see that the cyclic scheme can produce a high score among all four performance metrics, but since these scores do not share common hyperparameters we need to choose which of them to prioritize. Due to our interest in capturing the non-linear trends, we prioritize the score from the selected set of Tetrahedron patterns as this has proven to be the greatest challenge for our model to capture. We recognize that this choice introduces a greater risk of overfitting since the data points within this evaluation set are partly included in the training set as well. This is especially alarming since the absence of weight decay allows for more overfitting in general. However, for the purpose of performing an accelerated search, we find it more important to increase the likelihood of discovering novel designs than to reduce the risk of getting false positive results. Since we retain the option to verify the properties of a given design through MD

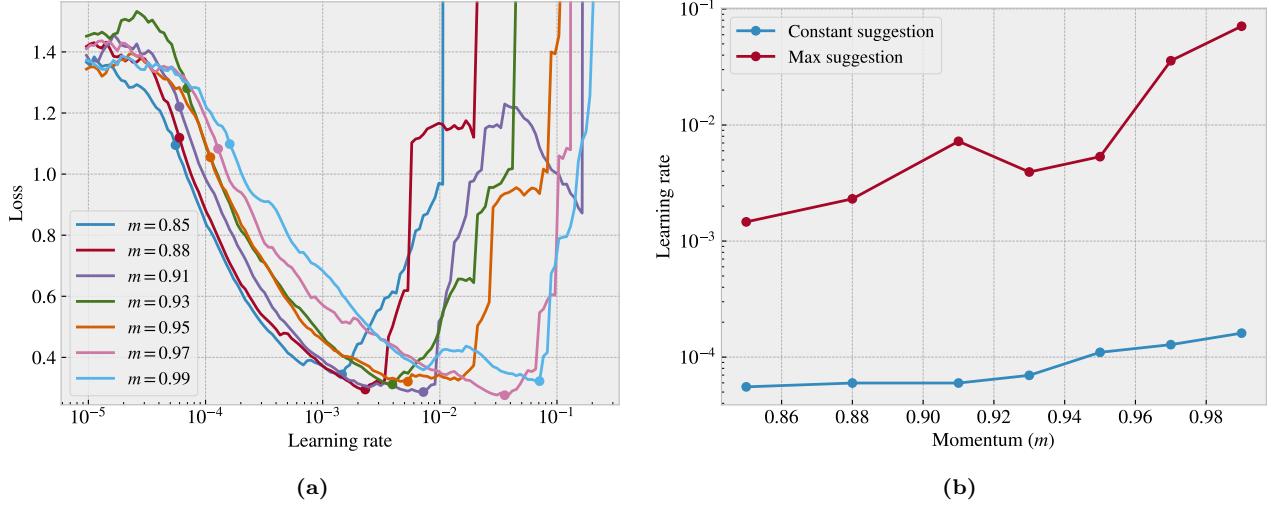


Figure 2.8: Learning rate range test for different momentum values m . (a) The validation loss for an exponentially increasing learning rate from 10^{-7} toward 10 with increments for each batch iteration during 1 epoch (yielding a total of 242 possible increments). As the curve diverges the test is halted. The dots on the validation loss vs. learning rate curve represent the steepest decline (at a lower learning rate) as an estimate for the constant learning scheme, and the minimum (at a higher learning rate) as an estimate for the maximum bound for the cyclic learning rate scheme. (b) The corresponding learning rate suggestion for the constant learning rate scheme and the maximum bound for the cyclic scheme respectively as a function of momentum choice.

simulations afterward, we do not have to rely on the machine learning prediction indefinitely. Thus we choose the cyclic trained model with low maximum momentum $m = 0.85$ and zero weight decay as our final model. On a final note, we also point out that our choice of hyperparameters corresponded to the edge of our grid search. Thus, it would have been natural to perform an extended search in that range, but due to time constraints and the belief that the potential gain from doing so was not significant, we decided to omit it.

Table 2.3: The best validation loss and R^2 scores from the momentum and weight grid search using the S32D12 model. We compare the scores from the constant and cycling learning rate and momentum scheme against the original scores from the S32D12 model in the architecture complexity grid search. The scores correspond to different hyperparameter choices that maximize each score respectively.

		Score [10^2]	Momentum	Weight decay
Validation loss	Original	2.124	0.9	0
	Constant	2.038	0.97	10^{-4}
	Cyclic	1.761	0.85	10^{-4}
Validation R^2	Original	98.16	0.9	0
	Constant	98.23	0.93	0
	Cyclic	98.31	0.93	10^{-5}
Tetrahedron R^2	Original	87.07	0.9	0
	Constant	87.31	0.85	10^{-5}
	Cyclic	88.12	0.85	0
Honeycomb R^2	Original	94.78	0.9	0
	Constant	95.67	0.93	0
	Cyclic	96.37	0.85	0

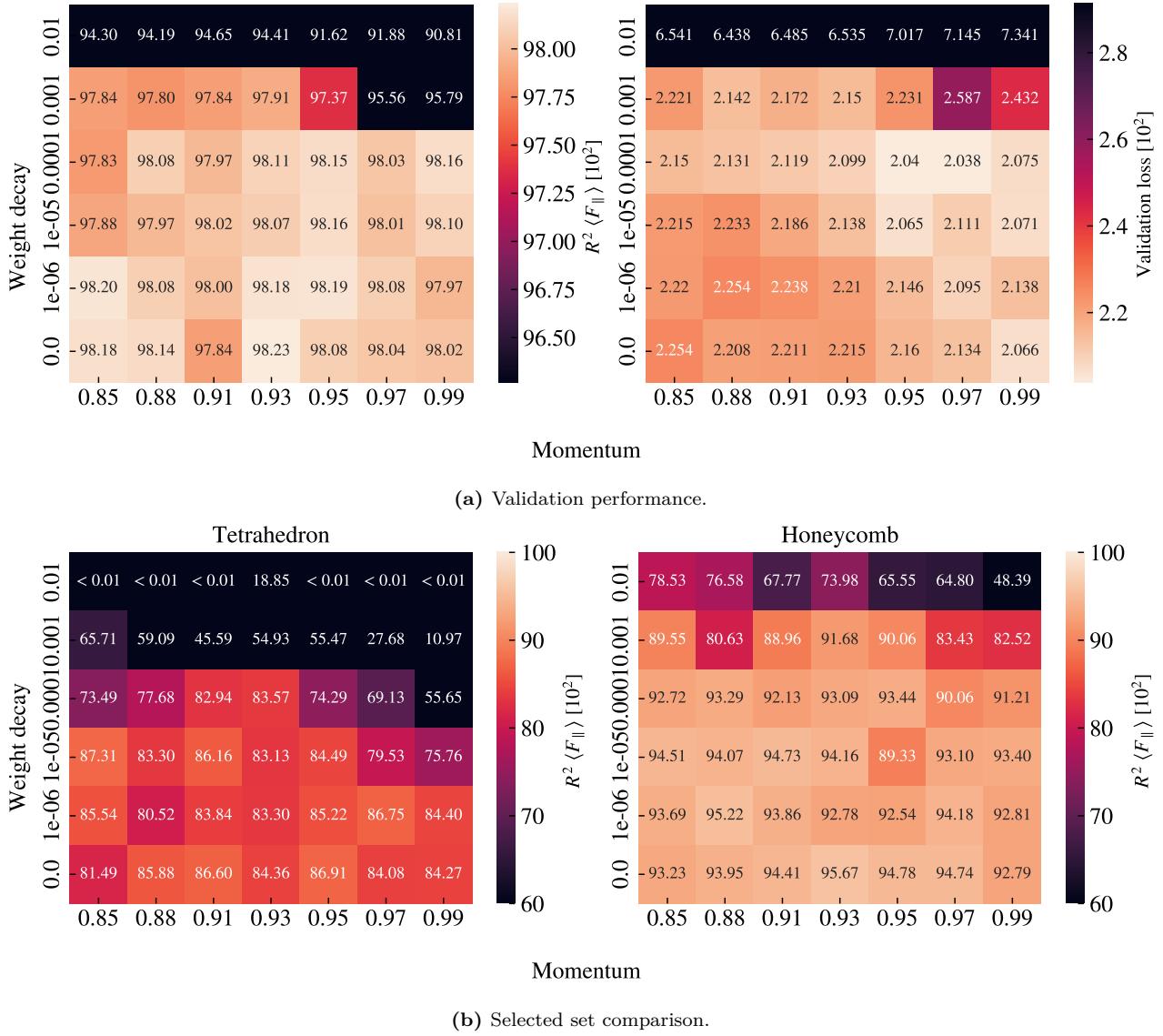


Figure 2.9: Momentum and weight decay grid search using a constant learning rate corresponding to the results from the learning rate range test in Fig. 2.8. (a) The friction mean R^2 validation score (left) and the validation loss (right). (b) The friction mean R^2 validation score for the selected set of Honeycomb (left) and Tetrahedron (right) patterns respectively, similar to that used in Fig. 2.7.

2.4.5 Model performance

From the hypertuning process, we settled on the S32D12 model trained by a cyclic scheme with a maximum momentum of 0.85, a maximum learning rate bound of 0.015 chosen accordingly to the learning range test in Fig. 2.8 and zero weight decay. The model contains 1.3×10^7 model parameters. The main performance metrics are shown in Table 2.4 where “Tetrahedron” and “Honeycomb” refer to the selected set scores. Although we have mainly considered the mean friction R^2 score during the hypertuning we find that the performance on the remaining parameters is reasonable as well. The validation set reveals a final R^2 score for the mean friction of $\sim 98\%$ and a rupture accuracy of $\sim 96\%$. Since the data only contains roughly 12% ruptures this should be compared to a score of 88% corresponding to simply predicting no ruptures at all. We find a considerably large relative error for the rupture strain of $\sim 13\%$. However, this error is lower for the Tetrahedron (5.9%) and Honeycomb (1.5%) sets. Hence, it is possible to infer that the high relative error in the validation set is due to some instances of very low rupture strains that might shift up the average value for the relative error.

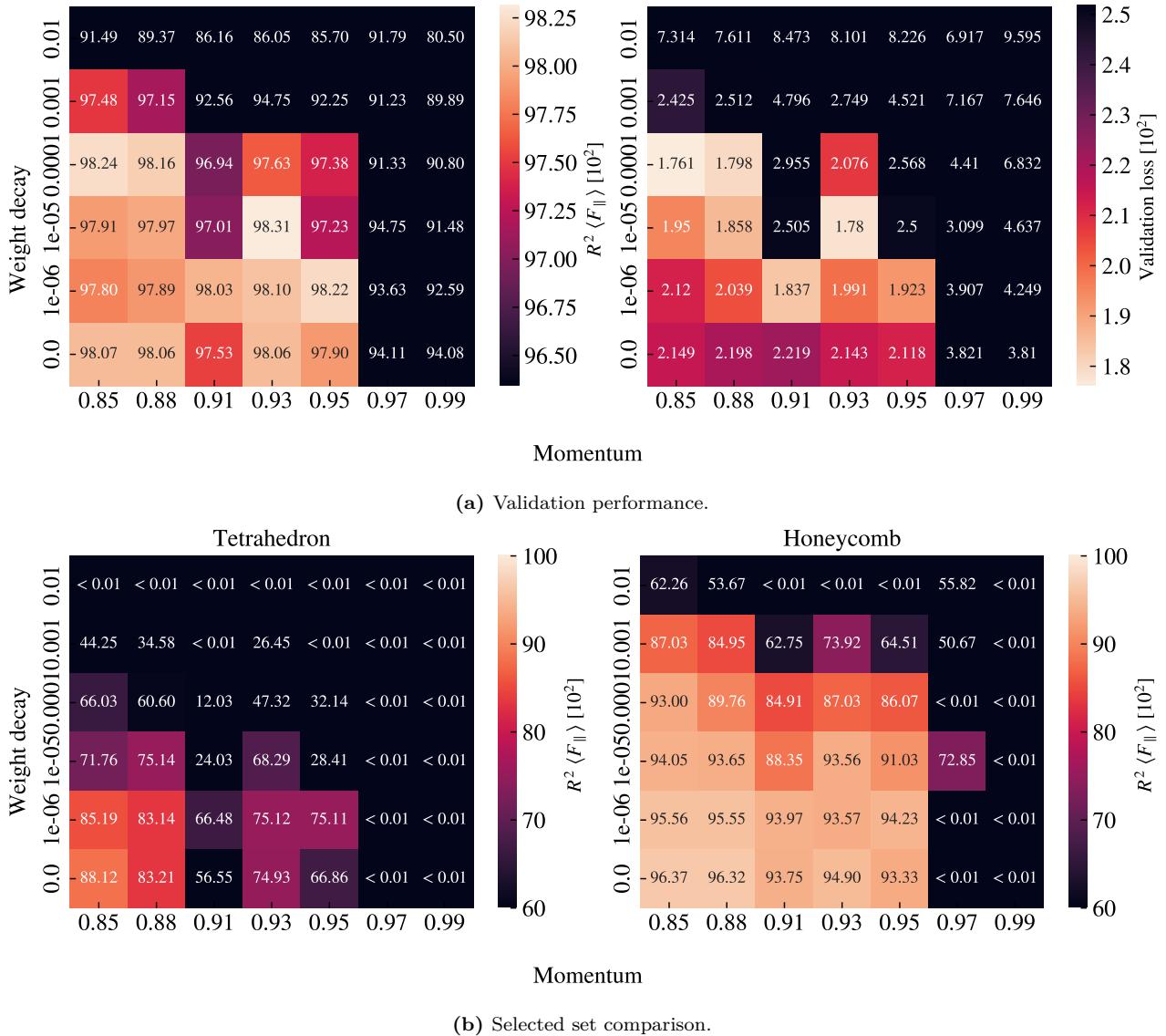


Figure 2.10: Momentum and weight decay grid search using a cyclic learning rate and cyclic momentum scheme. The learning rate maximum bound is chosen according to the learning rate range test in Fig. 2.8. The learning rate starts from a factor 1/20 of the maximum bound, increases toward the maximum bound during the first 30% of training and decreases toward a factor 10^{-4} of the maximum bound for the remaining 70% of training. This is done by following a cosine curve. The momentum performs an inverse cycling with the lowest momentum of 0.80 at the highest learning rate and a peak in momentum, corresponding to the values in the grid search, at the lowest learning rate. (a) The friction mean R^2 validation score (left) and the validation loss (right). (b) The friction mean R^2 validation score for the selected set of Honeycomb (left) and Tetrahedron (right) patterns respectively, similar to that used in Fig. 2.7.

Fig. 2.11 shows the mean friction, max friction and relative contact predictions for the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) patterns used in the pilot study. We note that these configurations are also partly contained in the training data, but this serves as a way of comparing the prediction quality estimated for the R^2 scores with a visual evaluation. Later on, we will evaluate a true test set based on the proposals from the accelerated search.

With the use of our final model, we evaluate the performance for the task of ranking the configurations by the properties of interest. That is, we go through all the configurations in the dataset, for the Tetrahedron, Honeycomb and Random walk respectively, calculate the properties of interest and sort the configurations

Table 2.4: Evaluation of the final model performance considering all the model outputs. Mean values are taken over the scores for each configuration within the categories: The validation set, the selected Tetrahedron set and the selected Honeycomb set.

	Loss [10 ²]	R^2 [10 ²]			Abs. [10 ²]	Rel. [10 ²]	Acc. [10 ²]
	Total	Mean F_f	Max F_f	Contact	Porosity	Rup. Strain	Rupture
Validation	2.1488	98.067	93.558	94.598	2.325	12.958	96.102
Tetrahedron	4.0328	88.662	85.836	64.683	1.207	5.880	99.762
Honeycomb	8.6867	96.627	89.696	97.171	1.040	1.483	99.111

accordingly. This is shown in Table 2.5 in comparison to the actual ranking in the dataset. Generally, we find that the ML model performs rather well in the ranking of the maximum properties getting the right configurations into the top 3 three, while it is performing a lot worse for the minimum friction property. This latter observation can be attributed to the fact that the precision needed for an accurate ranking among the minimum friction cases is a lot higher than for the remaining properties. This lack of precision is especially highlighted by the fact that the model predicts the 12th Random walk pattern (see Table 2.5c) to have negative mean friction, which is clearly outside the bound of reasonable values. For the maximum categories, we find that the model gives a slightly better ranking for the Tetrahedron and Honeycomb in comparison to the Random walk patterns. When considering the actual predicted property scores for the maximum properties we find that the model predictions are generally within a ~ 0.2 nN deviation in the top 5. This supports that our model can be used to perform an accelerated search of new configurations yielding a meaningful ranking of property scores.

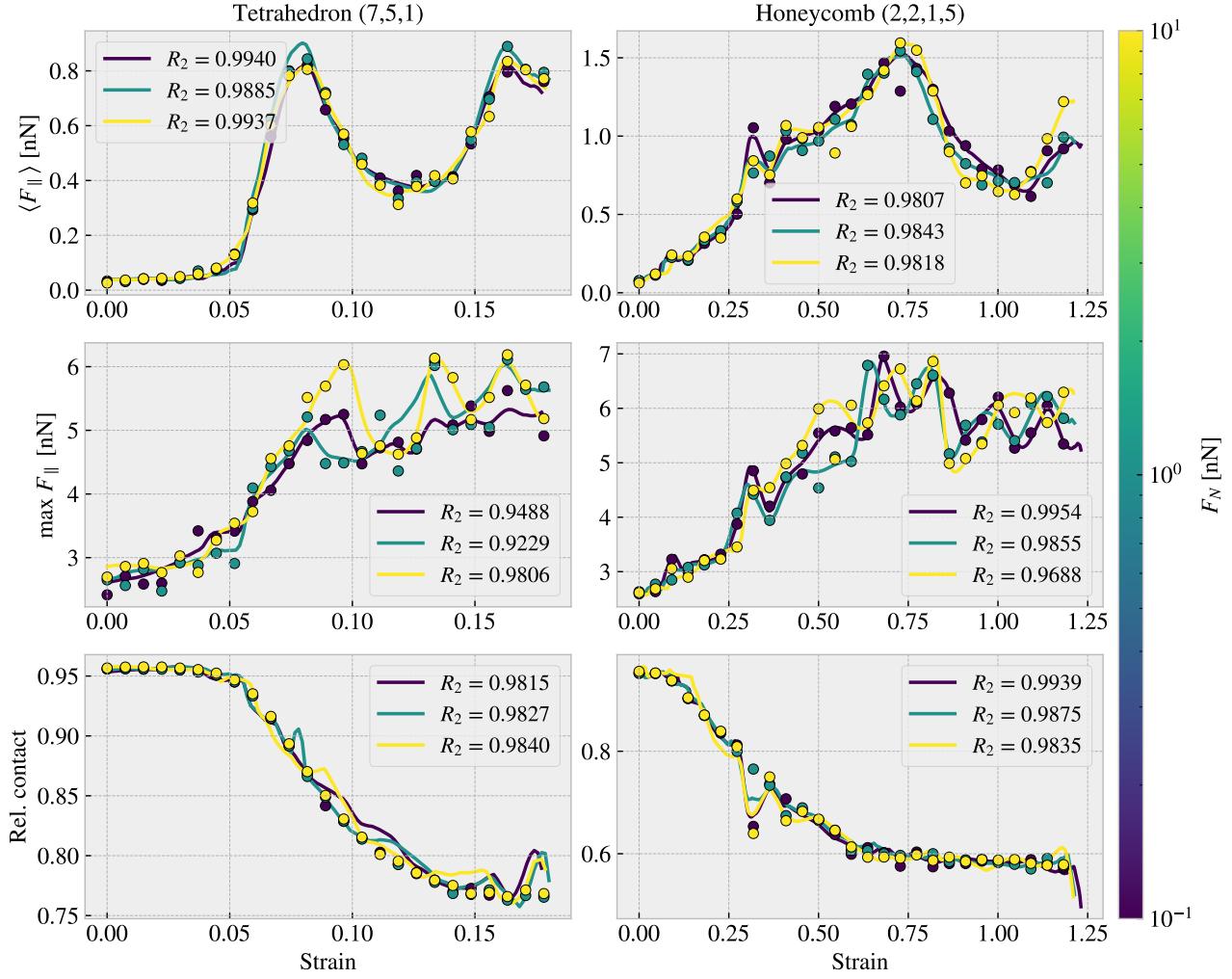


Figure 2.11: Visual evaluation of the final model predictions on the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) used in the pilot study, for the mean friction $\langle F_{\parallel} \rangle$, maximum friction $\max F_{\parallel}$ and the relative contact as a function of strain. The model predictions (solid line) are based on 10^3 data points in the strain range [0, 1.5], with the curve being cut off after the first rupture prediction. This is compared to the data points (dots). The color denotes the corresponding normal loads. The R^2 scores are shown for each prediction fit for each load value.

ML Rank	Data		ML		Data Rank
	Config	Value [nN]	Config	Value [nN]	
$\min F_{\text{fric}}$					
20	(3, 9, 4)	0.0067	(3, 1, 2)	0.0041	5
5	(3, 1, 3)	0.0075	(1, 3, 4)	0.0049	11
6	(5, 3, 4)	0.0084	(1, 3, 3)	0.0066	6
21	(1, 7, 3)	0.0084	(3, 1, 4)	0.0066	8
1	(3, 1, 2)	0.0097	(3, 1, 3)	0.0078	2
$\max F_{\text{fric}}$					
1	(5, 3, 1)	1.5875	(5, 3, 1)	1.5920	1
2	(1, 3, 1)	1.4310	(1, 3, 1)	1.2739	2
4	(3, 1, 2)	1.0988	(9, 3, 1)	1.1162	4
3	(9, 3, 1)	1.0936	(3, 1, 2)	0.7819	3
5	(7, 5, 1)	0.7916	(7, 5, 1)	0.7740	5
$\max \Delta F_{\text{fric}}$					
1	(5, 3, 1)	1.5529	(5, 3, 1)	1.5578	1
2	(1, 3, 1)	1.3916	(1, 3, 1)	1.2331	2
4	(3, 1, 2)	1.0891	(9, 3, 1)	1.0807	4
3	(9, 3, 1)	1.0606	(3, 1, 2)	0.7778	3
5	(7, 5, 1)	0.7536	(7, 5, 1)	0.7399	5
$\max \text{drop}$					
1	(5, 3, 1)	0.8841	(5, 3, 1)	0.8603	1
2	(3, 5, 1)	0.4091	(3, 5, 1)	0.3722	2
4	(7, 5, 1)	0.3775	(1, 1, 1)	0.2879	5
5	(9, 7, 1)	0.2238	(7, 5, 1)	0.2478	3
3	(1, 1, 1)	0.1347	(9, 7, 1)	0.1302	4

(a) Tetrahedron ranking.

ML Rank	Data		ML		Data Rank
	Config	Value [nN]	Config	Value [nN]	
$\min F_{\text{fric}}$					
1	(2, 5, 1, 1)	0.0177	(2, 5, 1, 1)	0.0113	1
9	(2, 4, 5, 1)	0.0187	(2, 5, 5, 3)	0.0149	7
7	(2, 4, 1, 1)	0.0212	(2, 5, 5, 1)	0.0182	4
3	(2, 5, 5, 1)	0.0212	(2, 5, 3, 1)	0.0186	5
4	(2, 5, 3, 1)	0.0226	(2, 4, 1, 3)	0.0198	15
$\max F_{\text{fric}}$					
1	(2, 1, 1, 1)	2.8903	(2, 1, 1, 1)	2.9171	1
2	(2, 1, 5, 3)	2.2824	(2, 1, 5, 3)	2.4004	2
6	(2, 1, 3, 1)	2.0818	(2, 1, 5, 1)	2.1060	5
4	(2, 1, 3, 3)	2.0313	(2, 1, 3, 3)	1.9458	4
3	(2, 1, 5, 1)	2.0164	(2, 4, 1, 1)	1.9381	6
$\max \Delta F_{\text{fric}}$					
1	(2, 1, 5, 3)	2.0234	(2, 1, 5, 3)	2.1675	1
2	(2, 1, 1, 1)	1.9528	(2, 1, 1, 1)	2.0809	2
3	(2, 4, 1, 1)	1.8184	(2, 4, 1, 1)	1.9157	3
4	(2, 1, 3, 3)	1.7645	(2, 1, 3, 3)	1.6968	4
5	(2, 4, 1, 3)	1.4614	(2, 4, 1, 3)	1.5612	5
$\max \text{drop}$					
1	(2, 3, 3, 3)	1.2785	(2, 3, 3, 3)	1.3642	1
2	(2, 1, 3, 1)	1.1046	(2, 1, 3, 1)	0.9837	2
3	(2, 3, 3, 5)	0.8947	(2, 3, 3, 5)	0.9803	3
4	(2, 1, 5, 3)	0.8638	(2, 1, 5, 3)	0.9556	4
13	(2, 5, 1, 1)	0.8468	(2, 4, 5, 3)	0.8999	8

(b) Honeycomb ranking.

Table 2.5: (Table continues on the next page)

ML Rank	Data		ML		Data Rank
	Config	Value [nN]	Config	Value [nN]	
min F_{fric}					
1	12	0.0024	12	-0.0011	1
24	76	0.0040	06	0.0036	27
6	13	0.0055	14	0.0074	23
31	08	0.0065	05	0.0082	19
26	07	0.0069	63	0.0085	57
max F_{fric}					
3	96	0.5758	99	0.5155	2
1	99	0.5316	98	0.4708	3
2	98	0.4478	96	0.4356	1
4	97	0.3624	97	0.3503	4
11	58	0.3410	55	0.2817	7
max ΔF_{fric}					
3	96	0.5448	99	0.4669	2
1	99	0.4769	98	0.4314	3
2	98	0.4085	96	0.4128	1
4	97	0.3268	97	0.3080	4
78	57	0.2978	55	0.2542	7
max drop					
3	01	0.1818	00	0.1883	3
2	96	0.1733	96	0.1654	2
1	00	0.1590	01	0.1532	1
11	37	0.1022	04	0.0591	8
28	34	0.0879	56	0.0552	20

(c) Random walk ranking.

Table 2.5: Ranking of the dataset according to the four properties of interest using the final machine learning (ML) model for the Tetrahedron (a), Honeycomb (b) and Random walk (c) patterns in the dataset respectively. The ranking is shown in descending order for each section of rows corresponding to the four properties of interest. The left side of the vertical center line denotes the true data ranking showing the top 5 scores in descending order (the top row shows rank 1 and the bottom row shows rank 5). The outermost left column (ML rank) then denotes the corresponding ranks given by the ML model. The right side of the vertical center line shows the top 5 ranking given by the ML model for which the outermost right column shows the corresponding true data ranks. If the model gets the top 5 ranking right both the outermost left and right columns show 1, 2, 3, 4, 5 in descending order.

2.5 Accelerated Search

From Sec. 2.4 we have found promising results that we can use our machine learning model to predict the friction behavior. This enables us to omit the MD simulations in the evaluation process and perform an accelerated search through new configurations. We will use the friction properties of interest as our main metrics for optimization. We approach the accelerated search by two different methods:

1. Using the generative algorithms developed for the creation of the Tetrahedron, Honeycomb and Random walk patterns, we create an extended dataset and evaluate the performance using the ML model.
2. Using a genetic algorithm method we perturb (mutate) the configurations and optimize for the maximum drop property using the ML model to evaluate the fitness function.

2.5.1 Patteren generation search

We utilize the pattern generators developed in ?? to create an extended dataset for our search. For the Tetrahedron and Honeycomb patterns, the increments of the parameters will eventually lead to the main pattern

structures exceeding the size of the sheet. Thus, we can essentially perform a full search “maxing out” the parameters of these patterns. We estimate that this is done with the max parameters, $(60, 60, 30)$ for the Tetrahedron, and $(30, 30, 30, 60)$ for the Honeycomb pattern. We use a random reference position and regenerate each unique parameter 10 times to explore translational effects. This gives in total 1.35×10^5 configurations for the Tetrahedron pattern and 2.025×10^6 for the Honeycomb pattern. For the Random walk generator, we perform a Monte Carlo sampling. That is, in each sample we draw the scalar values, either from a uniform (U) or logarithmic uniform (LU) distribution as follows.

$$\begin{array}{lll} \text{Num. walks} \sim U[1, 30] & \text{Max. steps} \sim U[1, 30] & \text{Min. dis.} \sim U[0, 4] \\ \text{Bias direction} \sim U[0, 2\pi] & \text{Bias. strength} \sim LU[0, 10] & p_{\text{stay}} \sim U[0, 1] \end{array}$$

Notice that we use a discrete distribution for the parameters requiring integers. For the binary parameters *connection*, *avoid invalid*, *RN6* and *grid start* we simply set the values by a 50–50 chance. The remaining parameters are kept constant at *periodic* set to true and *centering* set to false throughout the search. For the handling of clustering, we implement the extended repair algorithm such that the sheet is repaired by the least modifications approach rather than retrying the generation several times. Due to the extra computation time associated with the random walk and the repair algorithm, we only generate 10^4 configurations within this class. For the ML evaluation of the generated configurations we use a normal load of 5 nN and generate a friction-strain curve in the strain domain $[0, 2]$ using 100 uniformly spaced points. We compute the properties of interest and rank the configurations accordingly. The top candidate scores for each property are shown in Table 2.6 including a comparison to the original dataset top candidates (from Table 2.2). The random walk top five candidates are visualized in Fig. 2.12.

Table 2.6: Results for the accelerated search using the pattern generators. The top search candidates for each of the four properties of interest are shown in the left section (Search) regarding the Tetrahedron, Honeycomb and Random walk patterns respectively. The upper rows show the scores and the lower rows the associated names (parameters). The right section (Data) shows the corresponding scores from the best candidates within the dataset (from Table 2.2). All scores are given in units nN.

Scores	Search			Data		
	Tetrahedron	Honeycomb	Random walk	Tetrahedron	Honeycomb	Random walk
min F_{fric}	-0.062	-0.109	-0.061	0.0067	0.0177	0.0024
max F_{fric}	1.089	2.917	0.660	1.5875	2.8903	0.5758
max ΔF_{fric}	1.062	2.081	0.629	1.5529	2.0234	0.5448
max drop	0.277	1.250	0.269	0.8841	1.2785	0.1818
<hr/>						
Configs.	Tetrahedron	Honeycomb	Random walk	Tetrahedron	Honeycomb	Random walk
min F_{fric}	(13, 11, 14)	(14, 25, 7, 19)	<i>no name</i>	(3, 9, 4)	(2, 5, 1, 1)	12
max F_{fric}	(1, 3, 1)	(2, 1, 1, 1)	<i>no name</i>	(5, 3, 1)	(2, 1, 1, 1)	96
max ΔF_{fric}	(1, 3, 1)	(2, 1, 1, 1)	<i>no name</i>	(5, 3, 1)	(2, 1, 5, 3)	96
max drop	(1, 7, 1)	(3, 3, 5, 3)	<i>no name</i>	(5, 3, 1)	(2, 3, 3, 3)	01

First of all, we notice that the top candidates for the minimum friction are all predicted to have a negative friction value. This unphysical prediction aligns with the previous observations that our model does not have the required precision to yield accurate predictions for this property. Moreover, we can argue that pursuing the optimization for a low friction value will eventually exploit the weaknesses of the model as we reward an unphysical negative value. In order to resolve this problem it may be necessary to extend the training dataset and possibly include a physical constraint for positive friction values. However, by consulting the proposed minimum candidates we find that they all share the same feature of being sparsely cut. For the Random walk, we see this visually in Fig. 2.12, while for the Tetrahedron and Honeycomb patterns, this is evident from the configuration parameters shown in Table 2.6 where the parameters reveal a high spacing between the cuts. The porosity of the minimum friction top candidates are all rather low being 1.5%, 5.6%, and 1.6% for the Tetrahedron, Honeycomb and Random walk respectively. This further supports the idea that the Kirigami sheet can not readily be used to reduce friction within our system domain since the results point toward a non-cut sheet as the best minimum friction candidate.

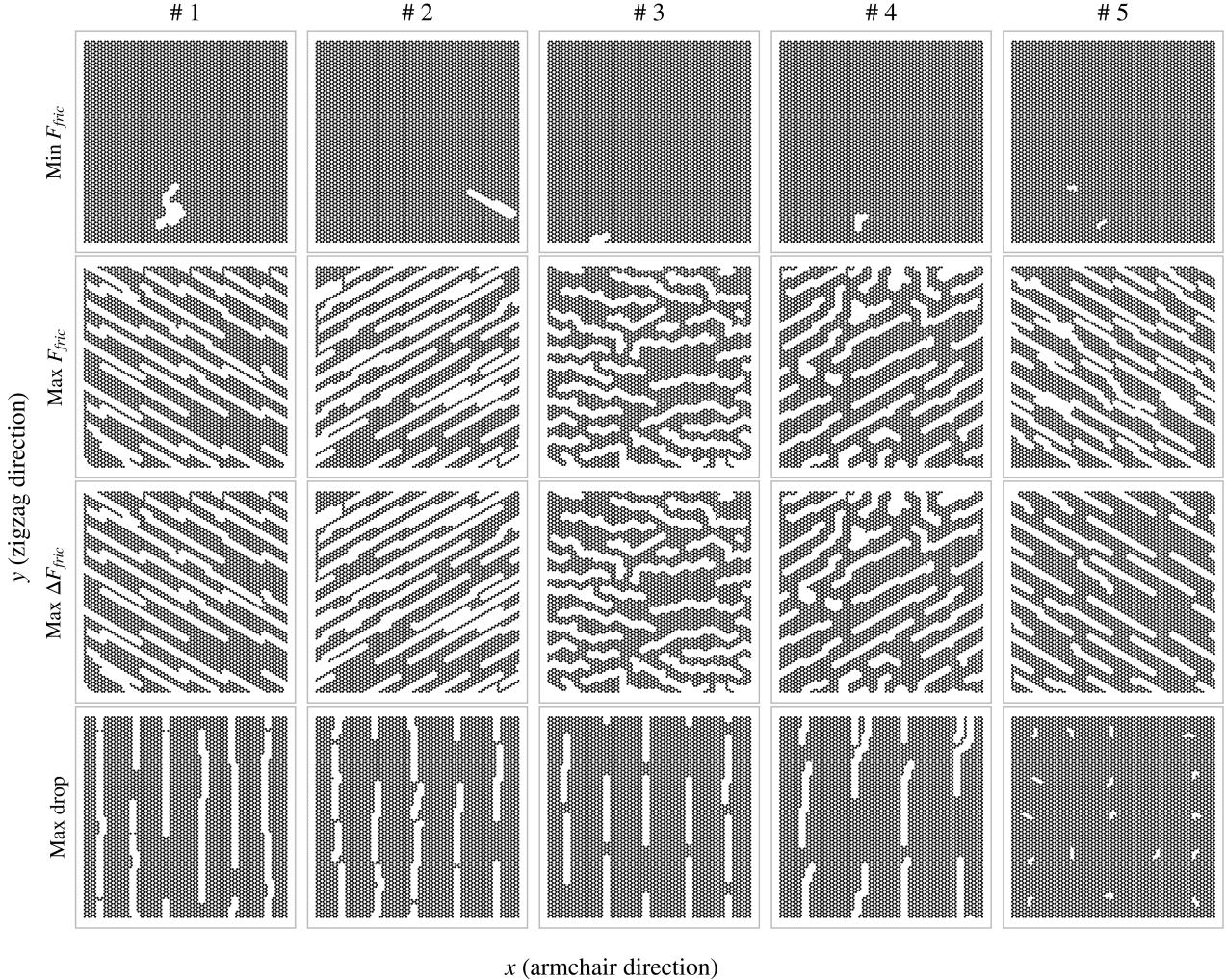


Figure 2.12: Top 5 candidates for the accelerated search using the Random walk generator. The rows denote the four properties of interest and the columns the top 5 candidates found in the search descending from left to right.

Among the remaining maximum properties, we find competing values for the Honeycomb and Random walk classes. However, the top-scoring values for the Honeycomb candidates correspond to configurations already within the original dataset, which is also the case for the Tetrahedron top candidates. The only difference is the randomized reference position making for a translated version of the pattern. When taking a closer look at the full ranking for each property it becomes apparent that the predictions are highly sensitive to the reference position parameter used for the Tetrahedron and Honeycomb pattern. Since we repeated each pattern parameter 10 times with a random reference position, we initially expected to get a ranking in sets of 10. However, the ranking only shows contiguous appearing sets in the range of 1–5 which points toward a dependency on pattern translation. Hence we investigate this further by evaluating the scores for a systematic change of the reference position. We generally find the max drop parameter to give the highest variation and thus we show the max drop scores for the max drop top candidates: Tetrahedron (1, 7, 1), (5, 3, 1) and Honeycomb (3, 3, 5, 3), (2, 3, 3, 3) in Fig. 2.13. The results show that the max drop property prediction varies drastically with the translation of these patterns. The emerging question is then whether this is grounded in a physical phenomenon or simply a deficiency in the model. Even though the patterns are periodic in the x-y-plane, with a period according to the unique number of translations shown in Fig. 2.13, the translation will determine the specific configuration of the edge. Previous studies of static friction and stick-slip behavior point to the importance of edge effects [26], and thus for a sheet where the atoms on the $\pm x$ free edge constitute about 2.5% of the inner sheet atom count, it is not unreasonable

that the translation might result in a significantly different outcome. In that case, the search through reference positions highlights that the translation can be key to optimizing for certain properties. However, the results might also indicate that the model is either overfitted or that we simply did not provide enough data to reach a generalization of the complex physical behavior of the system. The sensible way forward to unravel this would be to perform MD simulations for additional translational variants of the same configurations to investigate for any physical edge dependencies or otherwise strengthen the model by this data. We leave this suggestion for another study. When considering some of the friction-strain curves corresponding to the result in Fig. 2.13 we also find that the prediction of the rupture point plays an important role in the max drop property score. Since the rupture is often predicted on a descending part of the curve any variation to the rupture strain will affect the max drop property quite significantly.

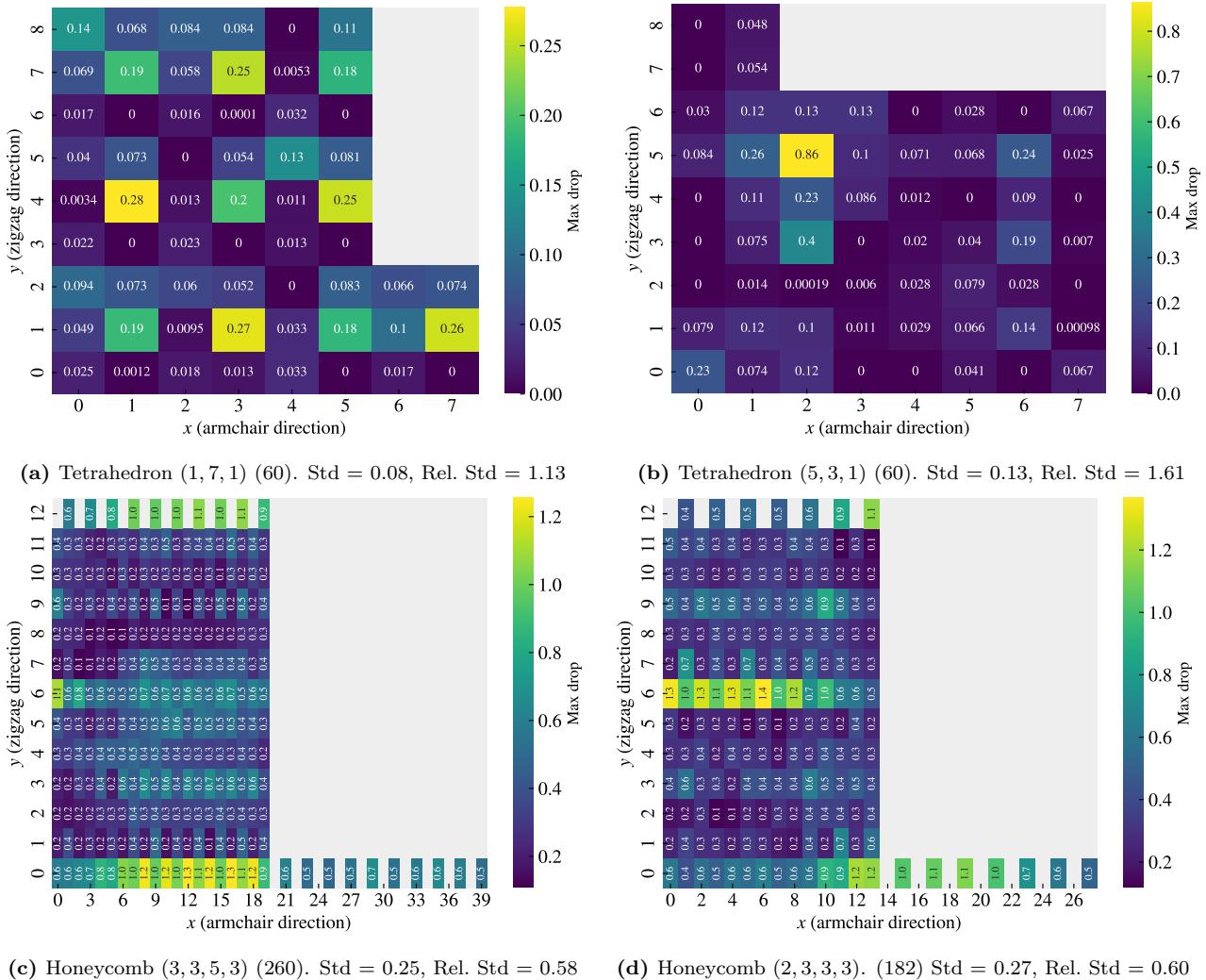


Figure 2.13: Prediction of the max drop property for selected patterns using the machine learning model for all unique reference positions. The heatmap and the annotated text denote the property score with respect to the reference position x and y coordinate. Panel (a)-(d) shows the results from various patterns which correspond to the candidates for the max drop property in the accelerated pattern search and the dataset respectively. The sub-caption states the pattern parameters, the number of total unique reference positions in parenthesis, the standard deviation and the standard deviation relative to the mean score.

In order to get more insight into the generalization of the model, we evaluate the performance on a true test set. We use the 20 configurations given by the top 5 Random walk candidates for each of the properties of interest shown in Fig. 2.12. We calculate the ground truth using MD simulations by sampling 30 pseudo uniform strain values within the rupture strain and by the use of a normal load of 5nN. Unfortunately, the

test set reveals a substantially worse performance than the validation scores reported so far. It shows a loss of 2.13, which is two orders of magnitude higher than the validation scores, an average absolute error for the mean friction of 0.14 nN and a rupture accuracy of 70 %. The corresponding mean friction average R^2 score is negative which indicates that our model performs worse than simply guessing the true data mean. This reveals that our model is not generalized enough to provide accurate predictions on the newly generated Random walk configurations. This can mainly be attributed to two reasons: 1) The test set data distribution is not similar to that of the training and validation data drawn from the original data set. 2) The considerations of the selected Tetrahedron and Honeycomb dataset, which overlapped with the training data, has led to an overfitting of the model in the hypertuning. In order to test the last hypothesis we went back to the beginning of the hypertuning process and chose the best model (C16D8) based purely on validation loss in the architecture complexity grid test Fig. 2.6. By using this model on the test set we find similar poor results which suggests that the test performance issue is not caused by the hypertuning process. Instead, it points to the fact that our original training data does not contain a generalized enough configuration distribution to accurately capture the full complexity of our system. This aligns with the high fluctuation in prediction value when translating the patterns. Thus we conclude that a machine learning approach might be feasible, given the promising validation scores, but that we need a bigger and more generalized dataset for a reliable prediction of new configurations.

2.5.2 Genetic algorithm search

Although our machine learning analysis indicates that the model is not generalized enough for an accurate prediction on new configurations, we carry out a short investigation on the use of a genetic algorithm based accelerated search. So far we have concluded that a minimization of the friction is not promising, and hence we discard this property for further study. We have also seen that the maximum style properties often share similar top candidates, and thus we choose to only investigate the max drop property associated with the aim of creating a negative friction coefficient in the coupled system.

In order to verify our implementation of the genetic algorithm described in ?? we consider a smaller 10×10 square lattice initially. We choose to evaluate the energy of a zero-temperature Ising model without any external field. That is, we consider the system described by the Hamiltonian

$$H = -J \sum_{\langle kl \rangle} s_k s_l, \quad (2.3)$$

where s denotes the value of each lattice site and the sum is running over all nearest neighbor pairs in the lattice. The sites take a binary value, either -1 or $+1$, meaning that the lowest energy (-200) is reached when all site values have an equal sign, all -1 or all $+1$. The highest energy (200) corresponds to a checkerboard pattern where each site has nearest neighbors with an opposite sign to its own. We consider periodic boundary conditions, meaning that the sites on the edge will be connected to the sites on the opposite edge. We create a small population of 10 individuals from a random noise initialization where each site has a probability of p being -1 and $p - 1$ being $+1$. Hence, the probability represents the average porosity of the configurations and we choose $p = 0.5$ as an unbiased choice. When considering the minimization of energy, the algorithm converges relatively fast (within tens of generations) to the best possible score. Thus, we consider the more challenging problem of maximizing energy which requires the configuration to reach the checkerboard pattern. Fig. 2.14 shows the score for a maximization of the energy using the genetic algorithm. We observe that the score converges faster initially, but that it eventually reaches the best score at generation 257. This indicates that the algorithm can handle an optimization problem with some level of spatial dependency. However, we note that some initializations were resulting in a saturating convergence, where two unsynchronized checkerboard patterns formed in distinct regions of the lattice. This leads to a local maximum because transitioning from two unsynchronized checkerboard patterns to a single synchronized pattern would necessitate a slight decrease in the score initially.

Moving on from the verification of the algorithm, we consider the optimization for the max drop property for the Kirigami patterns. We utilize the machine learning model to evaluate the friction-strain curve for each pattern and compute the corresponding max drop property value, similar to what we have done in Sec. 2.5.1. For the initialization of the population, we take a basis in the top candidates for the pattern generation search from Sec. 2.5.1. That is, we generate new configurations using the parameters that led to the highest score for the max drop property. We do this for the top candidates within the Tetrahedron, Honeycomb and Random walk

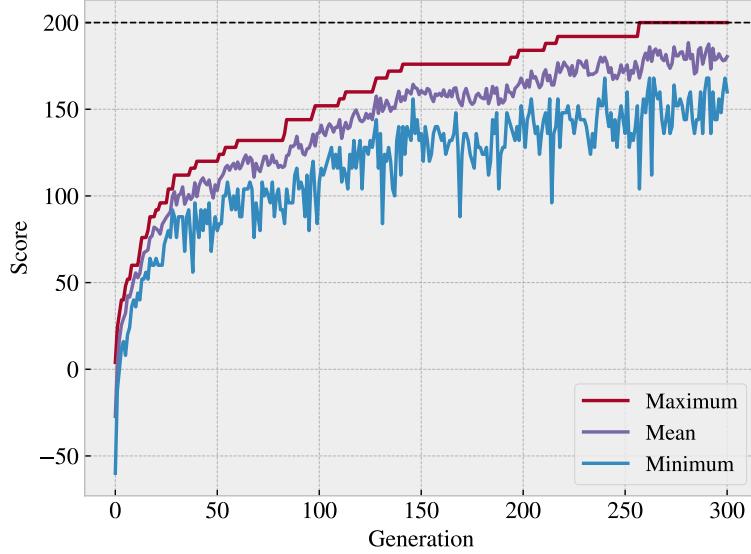


Figure 2.14: Optimization for the maximum Ising energy given by Eq. (2.3) for a 10×10 lattice using a population size of 10. The population was initialized with random noise having a probability of 0.5 being -1 and 0.5 being $+1$. The three curves indicate the minimum, mean, and maximum scores in the population. The best score indicated by the dotted line is reached at generation 257.

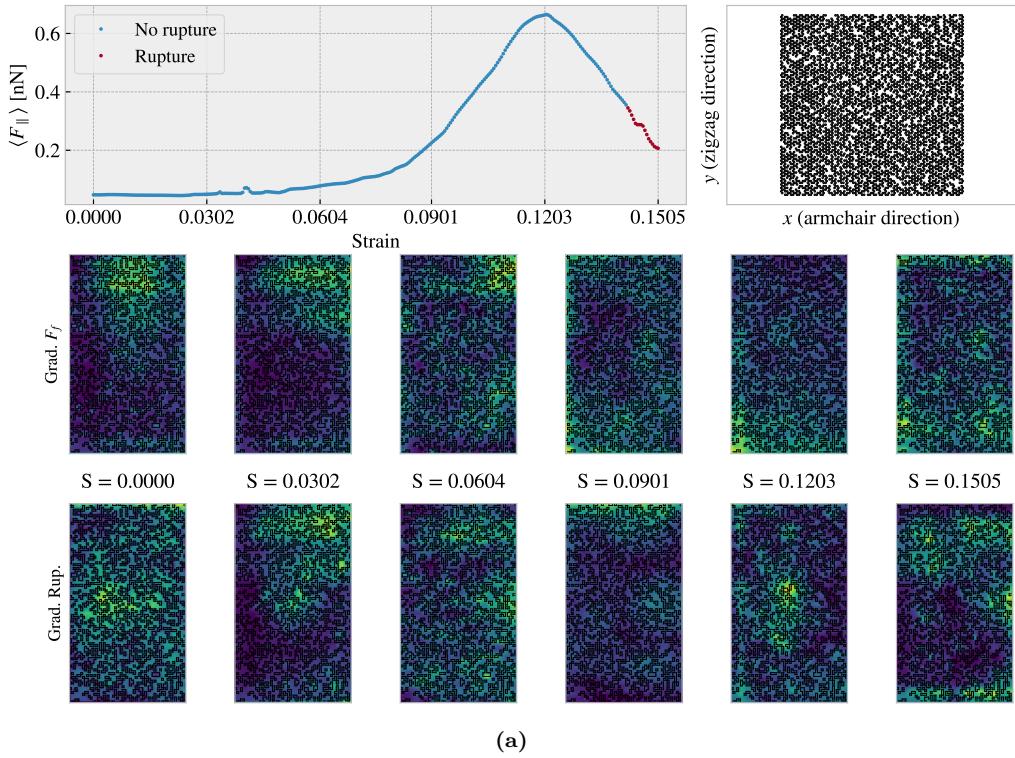
categories respectively. We generate a population of 100 configurations and run the search for 100 generations as we did not see much improvement for longer runs.

We find immediately that the Tetrahedron and Honeycomb search did not give any useful results. In both cases, the highest-scoring individual in the population at generation 0 was still the best candidate at the end of the search, even though the average score was rising initially. For the Random walk, we perform 5 runs for different population initializations. We select the top 5 max drop candidates from the Random walk pattern generation (seen in the bottom row in Fig. 2.12) and use their corresponding parameters for the initializations of the 5 populations. In 4 out of 5 runs, we find a similar result as seen for the Tetrahedon and Honeycomb patterns, meaning that only a single run provided a new pattern for the highest-scoring individual. The score of this individual was 0.240 nN which is only a small improvement from the otherwise best Random walk max drop score of 0.182 nN . However, from the other non-improving runs the initialization of the Random walk based population provided a top score of 0.345 nN which shows that we have better hopes of optimizing this property by simply generating more configurations from these parameters than running the genetic algorithm.

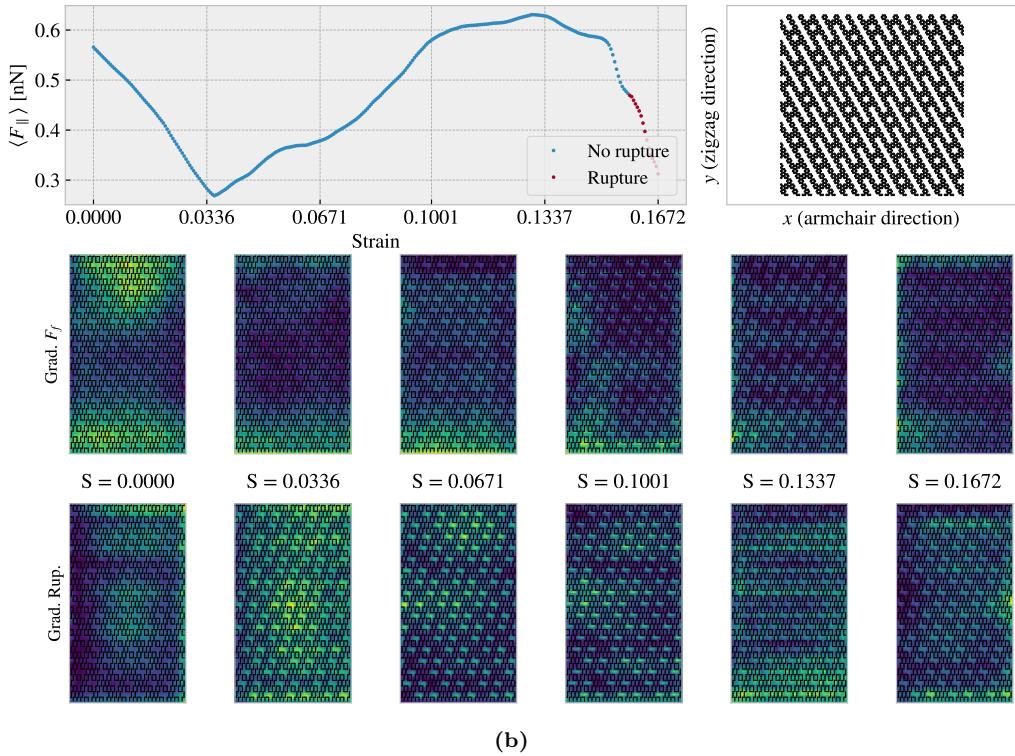
The fact that starting from an existing design did not give any useful results, we attempt to start from a population of random noise as well. We initialize one population with mixed porosities, having 20 individuals each for porosity $\{0.01, 0.05, 0.1, 0.2, 0.3\}$, and two populations based on a constant porosity of 0.25 and 0.5 respectively. This time the algorithm improved the highest-scoring individual throughout, but the final top scores are still not impressive. The mixed porosity start gave the highest score, being 0.299 nN . When inspecting the top five individuals in the population visually, they were all found rather similar to the starting configurations; they still looked like random noise. Thus, we do not find any significant signs that the genetic algorithm search can generate any higher-level pattern structures worth further investigating. We attribute this to the finding that the machine learning model is not reliable for predicting friction for newly generated configurations. However, we acknowledge that the quality of the genetic algorithm results could also be affected by an inadequate number of generations in the search. Nonetheless, given our concerns regarding the machine learning model, we decided not to pursue this any further.

As an attempt to get further insight into the model predictions, we use the Grad-CAM method to examine the top-scoring individuals from the genetic algorithm search. The result for the mixed porosity search is shown in Fig. 2.15a. For comparison, we included a similar examination for the top candidates in the pattern generation search, with respect to the max drop category, for the Tetrahedron, Honeycomb and Random walk, as shown in Fig. 2.15b to 2.15c. For the mixed porosity configuration, the Grad-CAM method highlights some areas in the noise as contributing more positively than others, but we do not find any obvious structure from this.

Regarding the Tetrahedron, Honeycomb, and Random Walk configurations with more organized patterns, we observe that the cuts are frequently highlighted. This finding gives some confidence to the notion that the model considers some of the relevant features in the pattern. Nonetheless, the variability of the results is too great to draw any firm conclusions. However, we notice that for certain strain values, the Grad-CAM reveals considerable “attention” toward the edge of the configuration. This especially relates to the top and bottom edge ($\pm y$). For instance, Fig. 2.15c shows a highlighting of the bottom edge at strain 0.396 for the Honeycomb pattern. Considering that the top and bottom of the configuration are not a true edge, since these are connected to the pull blocks in the simulation, this is a bit surprising. One interpretation is that the dissipation of energy associated with the thermostat in the pull blocks might be of importance. Even though these results should be taken carefully due to the instability of the model, we note this as a topic for further investigation.



(a)



(b)

Figure 2.15: (The figure continues on the next page)

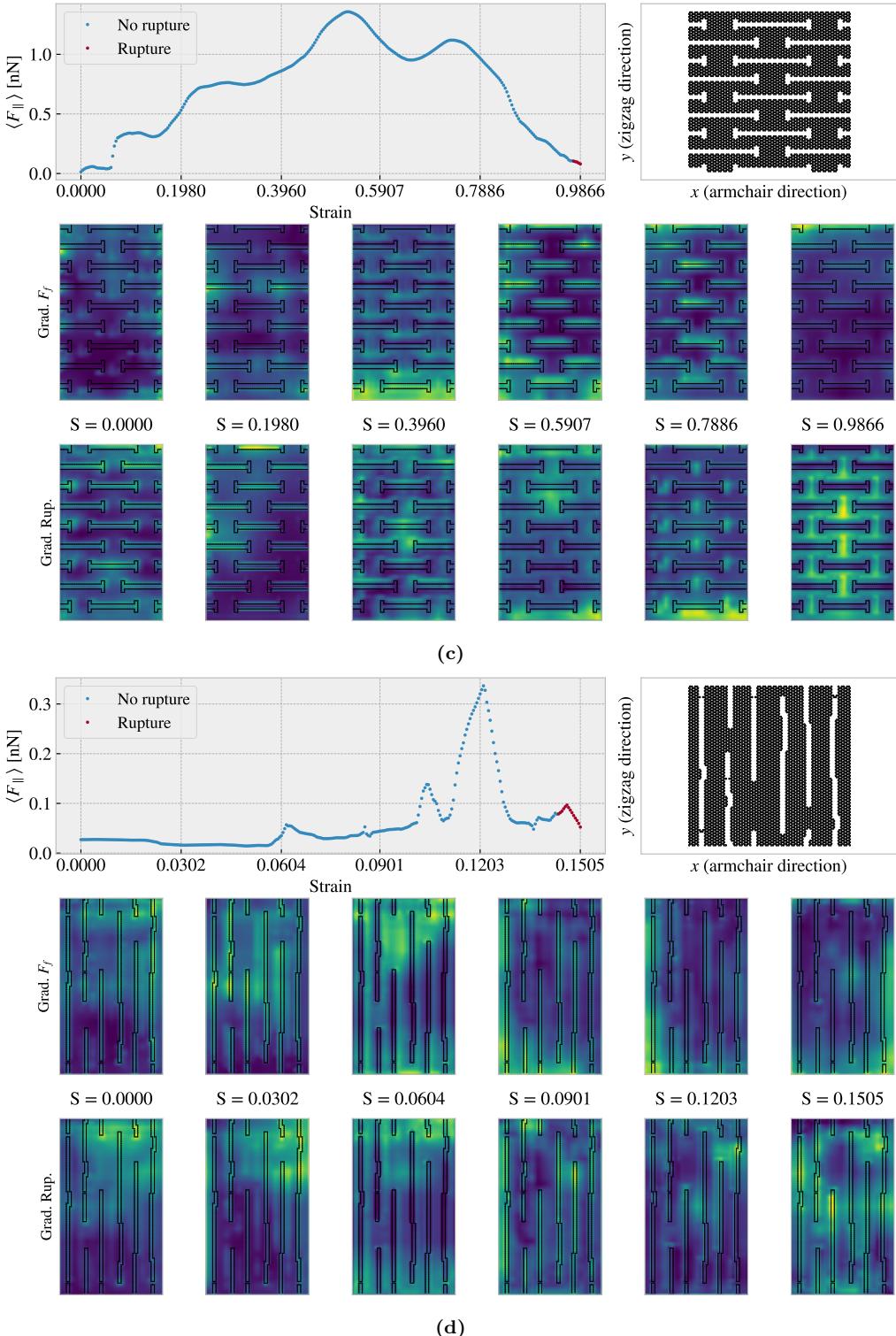


Figure 2.15: Grad-CAM analysis of selected Kirigami configurations. The top rows show the predicted friction-strain curve (left) with the rupture prediction indicated by the colors (blue: no rupture, red: rupture), and a visualization of the Kirigami graphene sheet on the hexagonal lattice (right). The remaining rows show the grad-CAM heatmaps with respect to the prediction of the friction (Grad. F_f) and rupture (Grad. rup) respectively, for various strain values corresponding to the x-axis ticks on the friction-strain curve. The four configurations displayed are: (a) The top-scoring individual from the genetic algorithm search using a mixed porosity start. (b) The Tetrahedron (1, 7, 1) with a reference point (1, 4) which gave the highest max drop score in the investigation of pattern translation in Fig. 2.13a. (c) The Honeycomb (3, 3, 5, 3) with a reference point (12, 0) which gave the highest max drop score in the investigation of pattern translation in Fig. 2.13c. (d) The highest scoring candidate for the max drop property in the Random walk pattern generation search shown in Fig. 2.12.

Appendices

Bibliography

- [1] E. Gnecco and E. Meyer, *Elements of friction theory and nanotribology* (Cambridge University Press, 2015).
- [2] Bhushan, “Introduction”, in *Introduction to tribology* (John Wiley & Sons, Ltd, 2013) Chap. 1.
- [3] H.-J. Kim and D.-E. Kim, “Nano-scale friction: a review”, *International Journal of Precision Engineering and Manufacturing* **10**, 141–151 (2009).
- [4] K. Holmberg and A. Erdemir, “Influence of tribology on global energy consumption, costs and emissions”, *Friction* **5**, 263–284 (2017).
- [5] B. Bhushan, “Gecko feet: natural hairy attachment systems for smart adhesion – mechanism, modeling and development of bio-inspired materials”, in *Nanotribology and nanomechanics: an introduction* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2008), pp. 1073–1134.
- [6] P. Z. Hanakata, E. D. Cubuk, D. K. Campbell, and H. S. Park, “Accelerated search and design of stretchable graphene kirigami using machine learning”, *Phys. Rev. Lett.* **121**, 255304 (2018).
- [7] P. Z. Hanakata, E. D. Cubuk, D. K. Campbell, and H. S. Park, “Forward and inverse design of kirigami via supervised autoencoder”, *Phys. Rev. Res.* **2**, 042006 (2020).
- [8] L.-K. Wan, Y.-X. Xue, J.-W. Jiang, and H. S. Park, “Machine learning accelerated search of the strongest graphene/h-bn interface with designed fracture properties”, *Journal of Applied Physics* **133**, 024302 (2023).
- [9] Y. Mao, Q. He, and X. Zhao, “Designing complex architectured materials with generative adversarial networks”, *Science Advances* **6**, eaaz4169 (2020).
- [10] Z. Yang, C.-H. Yu, and M. J. Buehler, “Deep learning model to predict complex stress and strain fields in hierarchical composites”, *Science Advances* **7**, eabd7416 (2021).
- [11] A. E. Forte, P. Z. Hanakata, L. Jin, E. Zari, A. Zareei, M. C. Fernandes, L. Sumner, J. Alvarez, and K. Bertoldi, “Inverse design of inflatable soft membranes through machine learning”, *Advanced Functional Materials* **32**, 2111610 (2022).
- [12] S. Chen, J. Chen, X. Zhang, Z.-Y. Li, and J. Li, “Kirigami/origami: unfolding the new regime of advanced 3D microfabrication/nanofabrication with “folding””, *Light: Science & Applications* **9**, 75 (2020).
- [13] OpenAI, *Dall-e2*, (2023) <https://openai.com/product/dall-e-2>.
- [14] Midjourney, *Midjourney*, (2023) <https://www.midjourney.com>.
- [15] Z. Deng, A. Smolyanitsky, Q. Li, X.-Q. Feng, and R. J. Cannara, “Adhesion-dependent negative friction coefficient on chemically modified graphite at the nanoscale”, *Nature Materials* **11**, 1032–1037 (2012).
- [16] B. Liu, J. Wang, S. Zhao, C. Qu, Y. Liu, L. Ma, Z. Zhang, K. Liu, Q. Zheng, and M. Ma, “Negative friction coefficient in microscale graphite/mica layered heterojunctions”, *Science Advances* **6**, eaaz6787 (2020).
- [17] D. Mandelli, W. Ouyang, O. Hod, and M. Urbakh, “Negative friction coefficients in superlubric graphite-hexagonal boron nitride heterojunctions”, *Phys. Rev. Lett.* **122**, 076102 (2019).
- [18] R. W. Liefferink, B. Weber, C. Coulais, and D. Bonn, “Geometric control of sliding friction”, *Extreme Mechanics Letters* **49**, 101475 (2021).
- [19] M. Metzsch, *Github repository*, <https://github.com/mikkelme/MastersThesis>.

- [20] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, "LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales", *Comp. Phys. Comm.* **271**, 108171 (2022).
- [21] E. M. Nordhagen, *LAMMPS simulator*, <https://github.com/evenmn/lammps-simulator>.
- [22] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dulak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerger, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment—a python library for working with atoms", *Journal of Physics: Condensed Matter* **29**, 273002 (2017).
- [23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: an imperative style, high-performance deep learning library", in *Advances in neural information processing systems 32* (Curran Associates, Inc., 2019), pp. 8024–8035.
- [24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv preprint arXiv:1409.1556 (2014).
- [25] Neurohive, *Vgg16 – convolutional network for classification and detection*, [Online; accessed 07/05/2023], (2018) <https://neurohive.io/en/popular-networks/vgg16/>.
- [26] N. Varini, A. Vanossi, R. Guerra, D. Mandelli, R. Capozza, and E. Tosatti, "Static friction scaling of physisorbed islands: the key is in the edge", *Nanoscale* **7**, 2093–2101 (2015).