

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

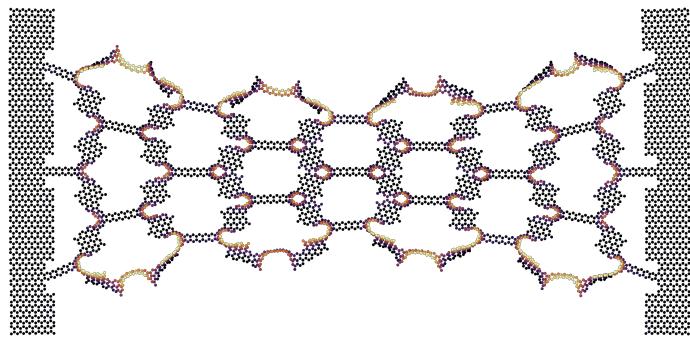
UNIVERSITY OF OSLO

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

Basic introduction

Various numerical models and experimental results propose different governing mechanisms for friction at the nanoscale.

More detailed background

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 in friction.

General problem

However, to the best of our knowledge, no previous studies have investigated the friction behavior of a nanoscale Kirigami graphene sheet under strain.

Summarize main result: “here we show”

Here we show that specific Kirigami designs yield a non-linear dependency between kinetic friction and the strain of the sheet.

General context

Using molecular dynamics simulation, 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. While this approach produced some promising results, we conclude that further improvements to the dataset are necessary in order to develop a reliable model.

Broader perspective

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.

353 words

Various numerical 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 in 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 simulation, 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. While this approach produced 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

Acknowledgments.

List of Symbols

F_N Normal force (normal load)

Acronyms

CNN Convolutional Neural Network. 17

GAN Generative Adversarial Networks. 2, 43

MD Molecular Dynamics. 1, 2, 3, 4, 9, 17, 25, 29, 32, 39, 40, 41

ML Machine Learning. 2, 26, 27, 28, 29, 30, 31, 42

MSE Mean Squared Error. 18

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Goals	3
1.3	Contributions	3
1.4	Thesis structure	4
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	12
2.4	Machine learning	17
2.4.1	Architecture	17
2.4.2	Data handling	18
2.4.2.1	Input	18
2.4.2.2	Output	18
2.4.2.3	Data augmentation	18
2.4.3	Loss	18
2.4.4	Hypertuning	19
2.4.5	Final model	25
2.5	Accelerated Search	29
2.5.1	Patteren generation search	29
2.5.2	Genetic algorithm search	33
3	Summary	39
3.1	Summary and conclusions	39
3.1.1	Designing an MD simulation	39
3.1.2	Genereting Kirigami patterns	40
3.1.3	Control friction using Kirigami	40
3.1.4	Capturing trends in friction data with machine learning	41
3.1.5	Accelerated search	42
3.1.6	Negative friction coefficient	42
3.2	Outlook	43
Appendices		45

Chapter 1

Introduction

1.1 Motivation

Friction is the force that prevents the relative motion of objects in contact. From our everyday life, we recognize it as the inherent resistance to sliding motion. Some surfaces appear slippery and some 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 [p. 5] [1]. 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” and 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, reducing friction is of great interest as this has advantages for energy efficiency. It has been reported that tribological problems have a significant potential for 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 are 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 geckos can produce large adhesive forces, they retain the ability to remove their 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 origin 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 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 architecture 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 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 accordingly to the most promising candidates, for instance, as done with the *genetic algorithm* based on mutation and crossing of the best candidates so far. 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 *reverse design*, where the model predicts a design from a set of 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. attributes 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 behavior 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 stretch 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 theory also take on negative values. The latter would constitute a rare property only found a few cases. These are mainly for the unloading phase of adhesive surfaces [15] or 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.

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 solutions, which is often beyond the capabilities of computational resources.

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) 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 framework 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 load and stretch for different Kirigami designs.
4. Develop and train a machine learning model to predict the MD simulation result and perform an accelerated search of new designs with the scope 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, but 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 it requires an extended dataset in order to produce reliable results for a search of new designs.

During our investigations, I have built three numerical tools, beyond the regular scripts associated with data analysis, which can be found 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 concerning the choice of sheet configuration, system size, simulation parameters and MD potentials, which makes it applicable for further studies within 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. 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 allow 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 used for this thesis have been originally developed for the purpose with the exception of the usage of libraries as mentioned above and commonly known 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 will formulate our research questions in the light of the friction theory.

In Part II, we begin by presenting our definition and setting up of the system in ?? . This includes the MD simulations and the generation of Kirigami designs. This is followed by a pilot study in ?? where we evaluate the simulation results for various physical conditions and compare a non-cut sheet to two different Kirigami designs. Further explorations of the Kirigami configurations are carried out in Chapter 2 which includes the generation of a dataset and the employment of machine learning and an accelerated search. 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 in ?? . Finally, we summarize our results and provide an outlook for further studies in Chapter 3. 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 towards 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 predicting 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 force did not prove to be dominant in the friction description we only sample 3 normal force 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 configurations are chosen to introduce as much variety as possible in order to contribute to a wide distribution of configurations in the data. Notice that not all submitted data points “make it” to the final dataset. This 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 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 completely 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

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. In the latter datasets, the reference point for the pattern 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.

had been 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.

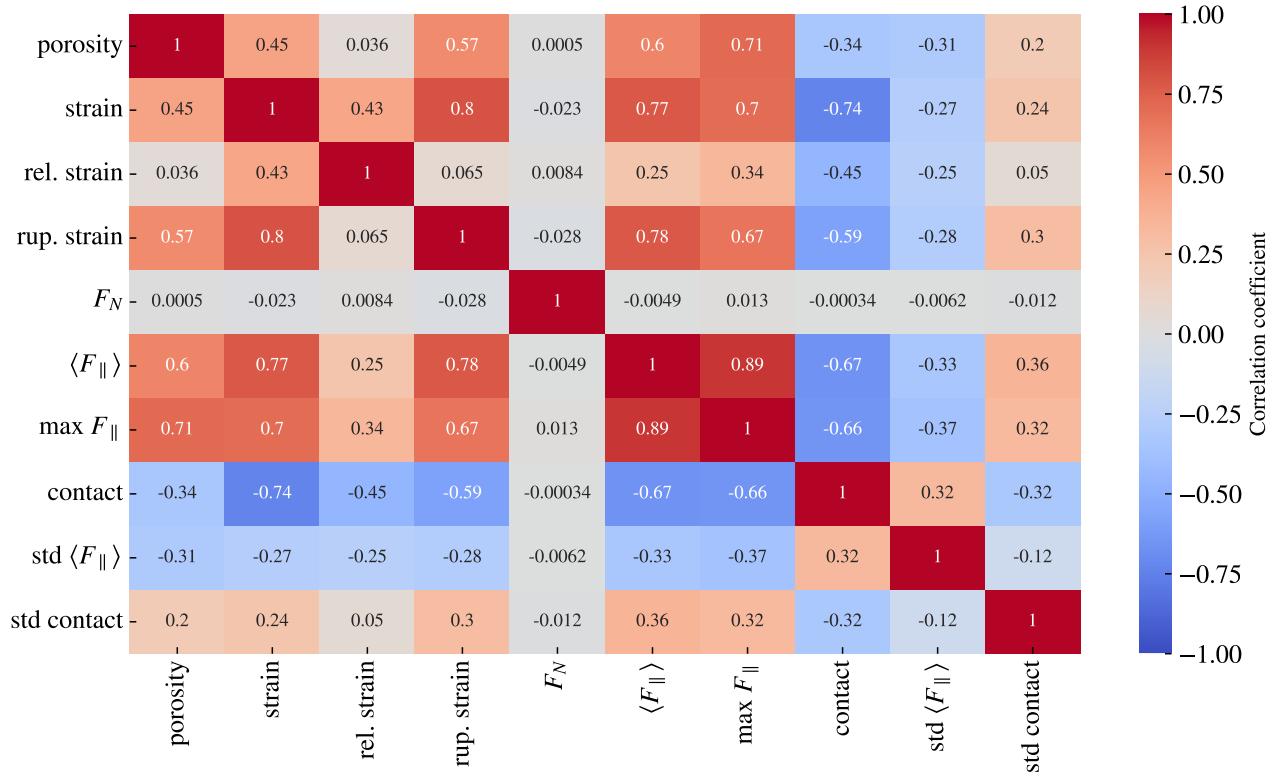


Figure 2.1: Pearson product-moment correlation coefficients Eq. (2.1) for the full dataset (see Table 2.1). Here the relative strain refers to the strain relative to the rupture strain.

From 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, the 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). From Fig. 2.1 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 chosen variable pairs on the axes. This allows us to visually identify certain 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.

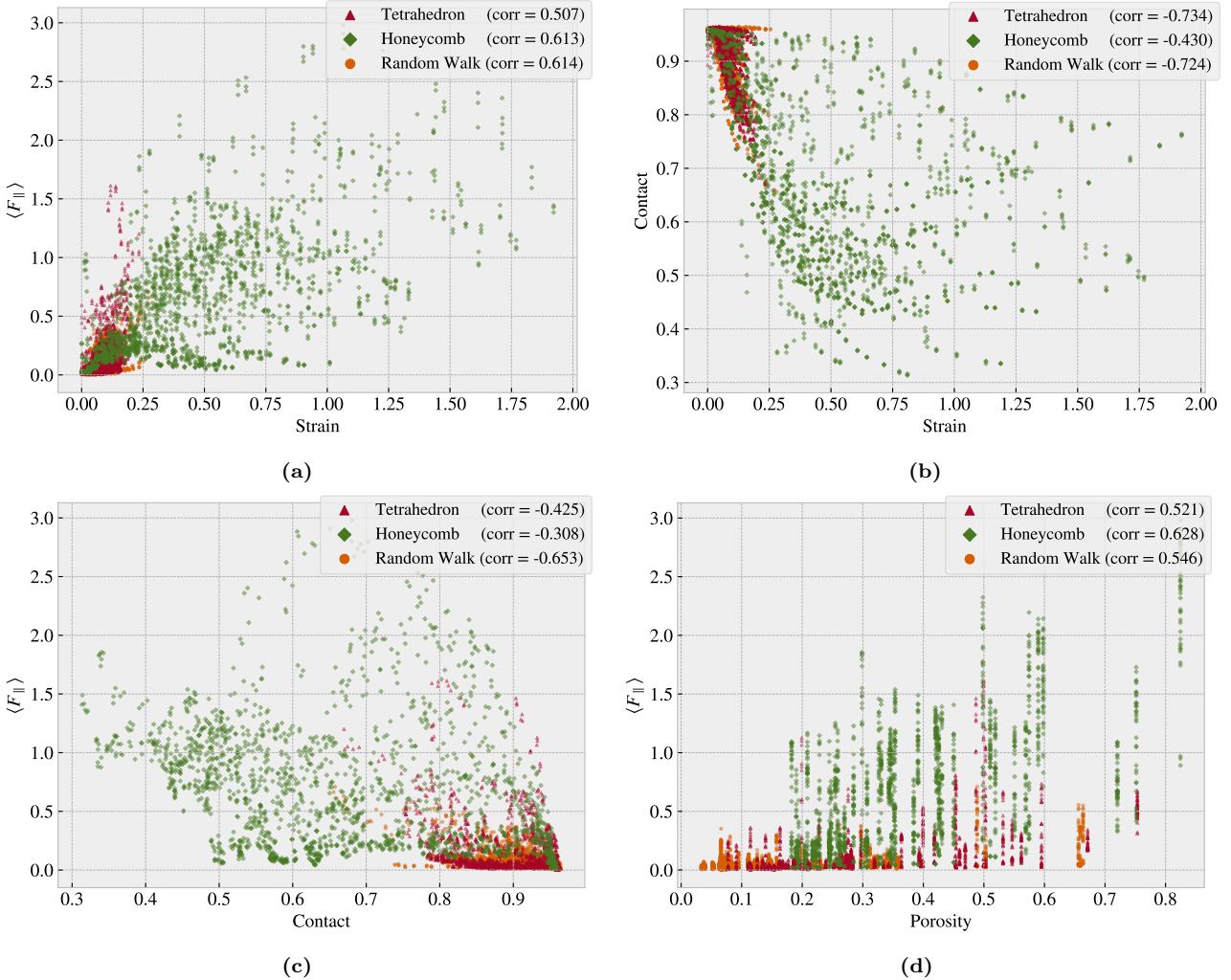


Figure 2.2: Scatter plot of various feature pairs in the data for the Tetrahedron, Honeycomb and Random Walk subsets respectively. (a) Mean friction vs. strain. (b) Relative contact vs. strain (c) Mean friction vs. relative contact (d) Mean friction vs. porosity.

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 force 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 $\mu = R\Delta F_f(\varepsilon)/\Delta\varepsilon$ 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. 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 $\Delta\varepsilon$, 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 drop 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* which serves as our indicator for a negative friction coefficient. In this evaluation, we do not guarantee a monotonic decrease of friction in the range of the biggest 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 biggest drop in for the friction-strain curve, we also consider the minimum, $\min F_{\text{fric}}$, the maximum, $\max F_{\text{fric}}$ and the maximum difference, $\max \Delta F_{\text{fric}} = \max F_{\text{fric}} - \min F_{\text{fric}}$. 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 of interest 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]	Tetrahedron (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

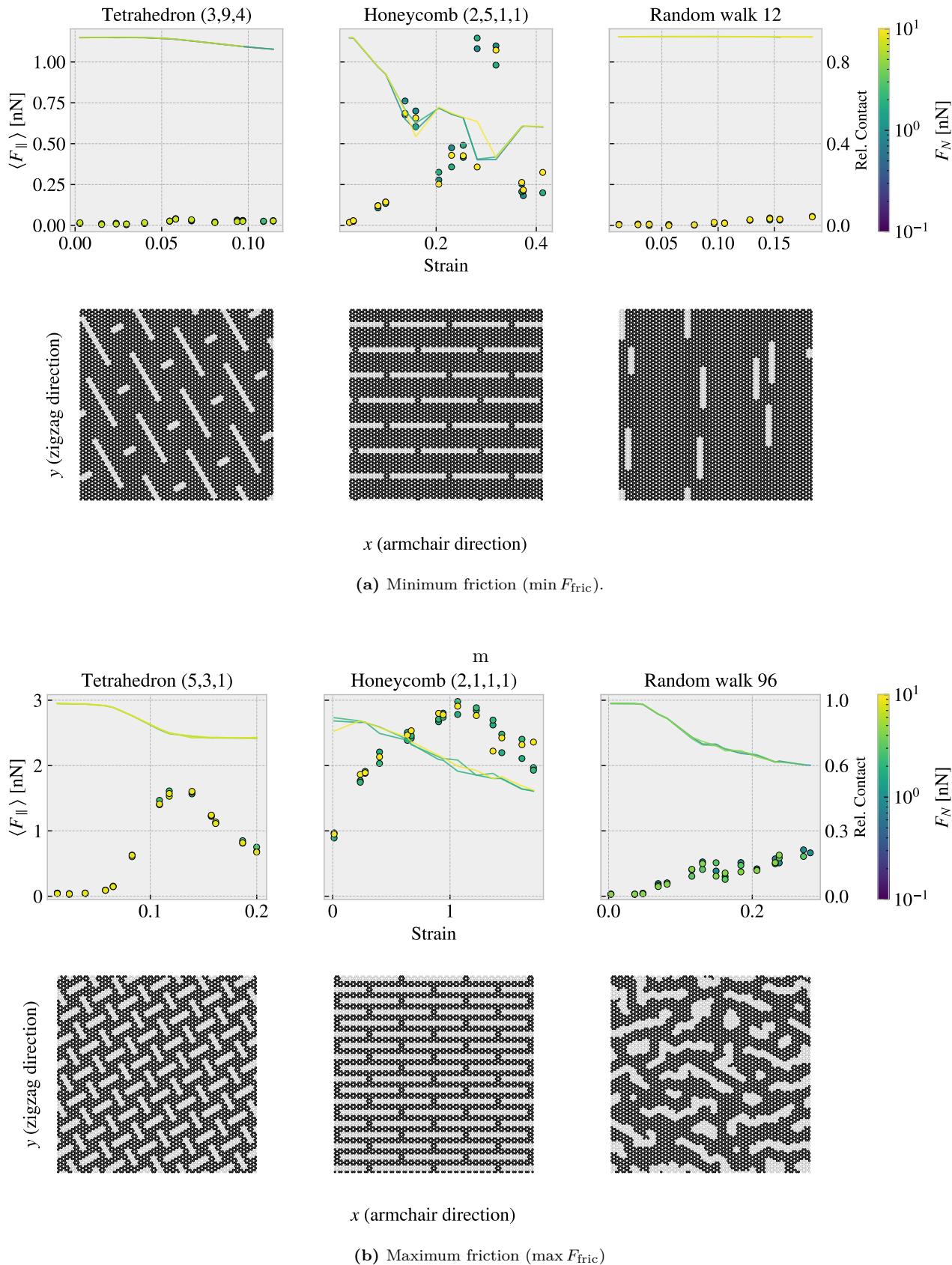
Random walk	Configuration	Strain	Value [nN]
$\min F_{\text{fric}}$	12	0.0562	0.0024
$\max F_{\text{fric}}$	96	0.2375	0.5758
$\max \Delta F_{\text{fric}}$	96	[0.0364, 0.2375]	0.5448
max drop	01	[0.0592, 0.1127]	0.1818

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 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 that of the Tetrahedron and Honeycomb structures. However, since these are found to be on a comparable order of magnitude we argue that these contribute with relevant information for the frictional dependency to the configuration space.

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 $\min F_{\text{fric}}$ top candidates (Fig. 2.3a) we find that the Random walk candidate has a rather low cut density (low porosity) and with 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 contact-strain curve being seemingly

constant. 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 relatively 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 are 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 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)

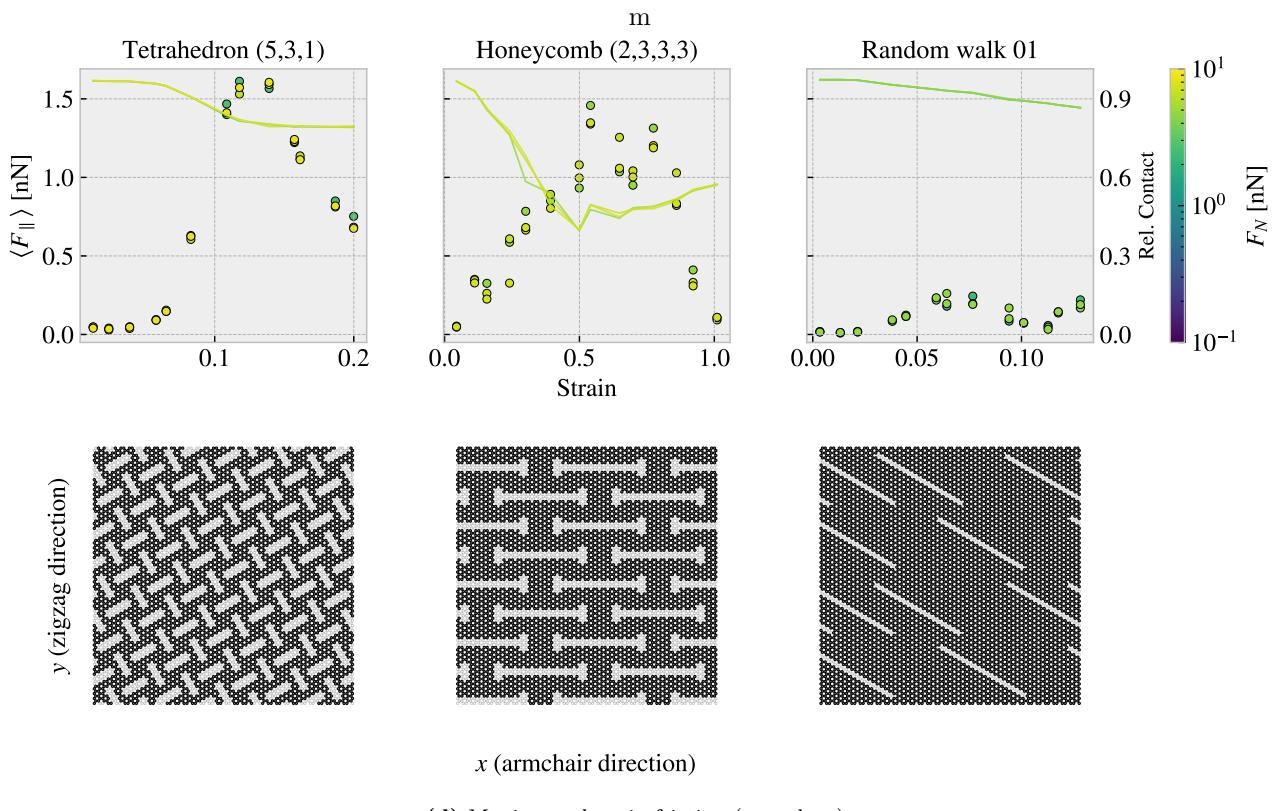
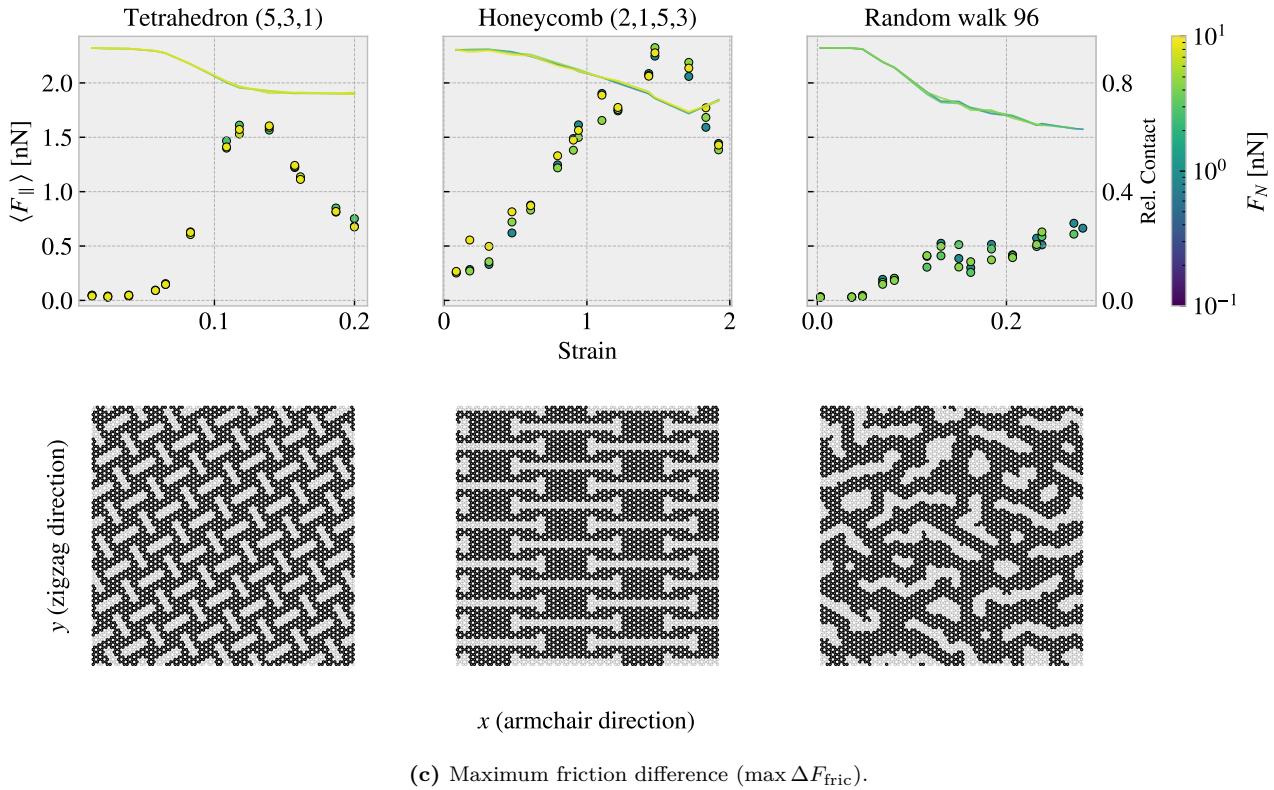


Figure 2.3: Illustration of the top candidates within each of the four properties of interest. The upper row shows the friction-strain curve with dotted points (with friction on the left y-axis) and the contact-strain curve with a solid line (with relative contact on the right y-axis). The lower row 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.

2.4 Machine learning

Given the MD-based dataset presented in the previous section: 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 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 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, 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 with the number of channels corresponding to the nodes in the first fully connected layer.

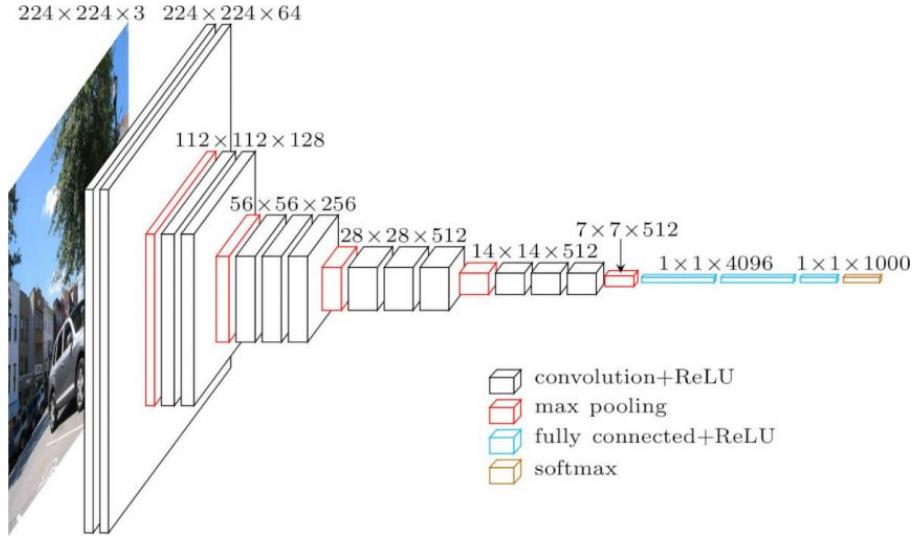


Figure 2.4: VGGNet-16. Source <https://neurohive.io/en/popular-networks/vgg16/>.

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 blocks: (Convolution \rightarrow Batch normalization \rightarrow ReLU \rightarrow Max pooling). Similarly, we define a fully connected block by two elements (Fully connected \rightarrow 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 process of determining a suiting complexity

for the architecture, we adopt the approach by Wan et al. [8] who used a “staircase” pattern for combining convolutional and fully connected blocks. By defining a starting number of channels S and 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 having the number of nodes decrease in a reverse pattern. For instance, the architecture $S4D8$ will take the form

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

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 configuration is given as a two-dimensional input by the 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 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 into the fully connected part of the network halfway in.

Both options utilize the same data, but the latter option is more directed towards independent processing of the data while the first makes for an intertwined use of the configuration, strain and load. We implemented both options but found immediately that option 1 was producing the most promising results during the initial test runs, 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, this 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, flip 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. We cannot use a reflection across the x-axis as the sheet is sliding in a positive y-direction. This 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 (MSE) ?? 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 variable

$$L_{tot} = \sum_v W_v \cdot L_v.$$

We choose the weights to be $1/2$ for the mean friction and $1/10$ for the remaining 5 variables, thus sharing the loss evenly for the remaining 50% of the weight. During the introductory phase of the training, 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 stuck to the values defined above.

2.4.4 Hypertuning

For the hypertuning we focus on architecture complexity, learning rate, momentum and weight decay. We will use the ADAM optimizer with the initial default values of $\beta_1 = 0.9$, $\beta_2 = 0.999$ and zero weight decay (we will change momentum β_1 and weight decay). We use a batch size of 32 and train the model for a maximum of 100 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 major 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 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 just a single epoch where a batch size of 32 yields a total of 242 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 .

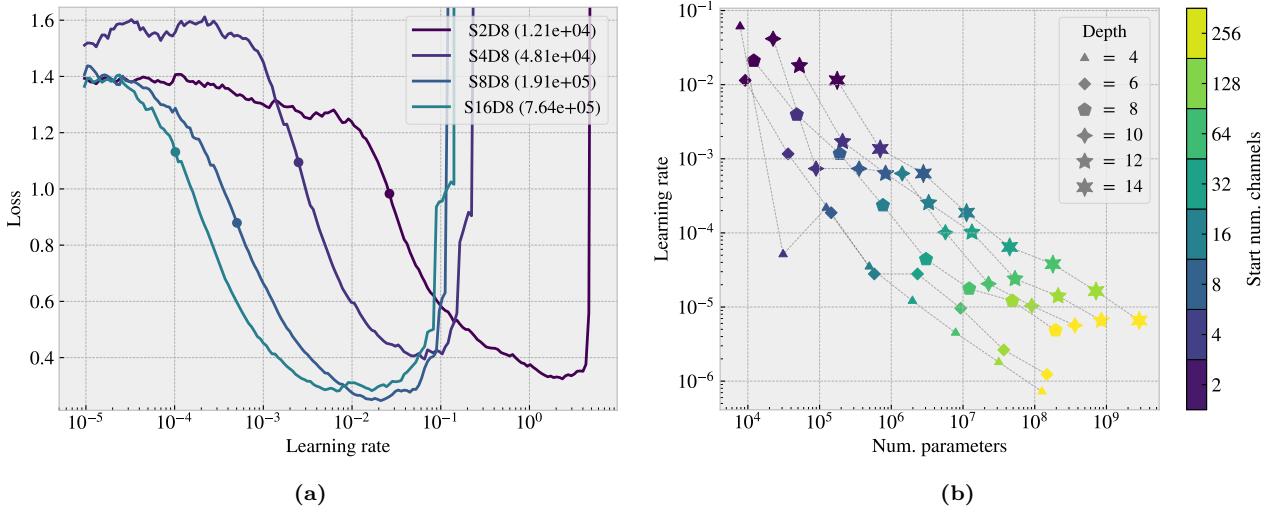


Figure 2.5: Learning rate range test for various model complexities. We increase the learning rate exponentially from 10^{-7} to 10 during one epoch corresponding to an exponent increment of roughly $1/30$ per batch iteration. (a) shows a few examples of the training loss history as a function of the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

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 strain curves which we immediately found to be the more difficult trend to capture. The selected evaluation is shown in Fig. 2.7. Note that these configurations already are 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.

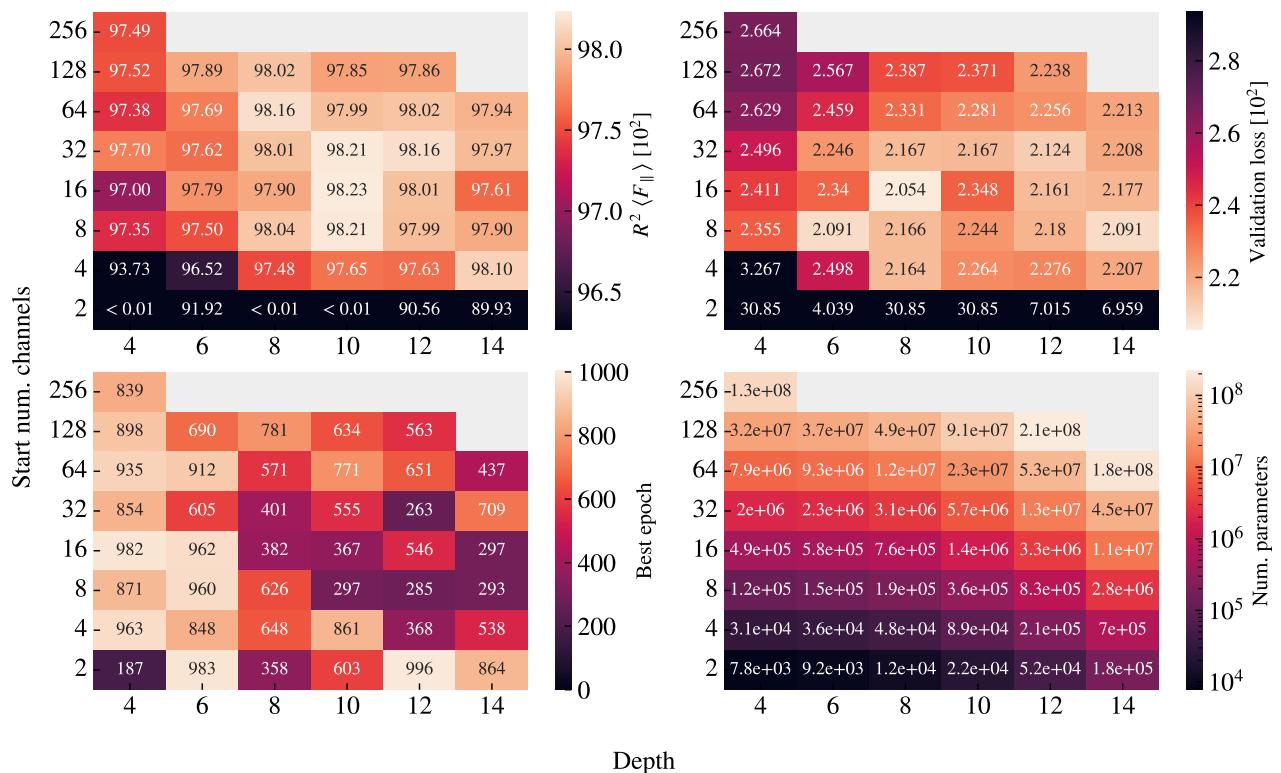


Figure 2.6: Architecture complexity grid search using a staircase-like VGG-net structure. The x-axis denotes the number of layers in the network (Depth) and the y-axis is the starting number of channels. See Eq. (2.2) for an example. For each architecture complexity, we evaluate the friction mean R^2 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).

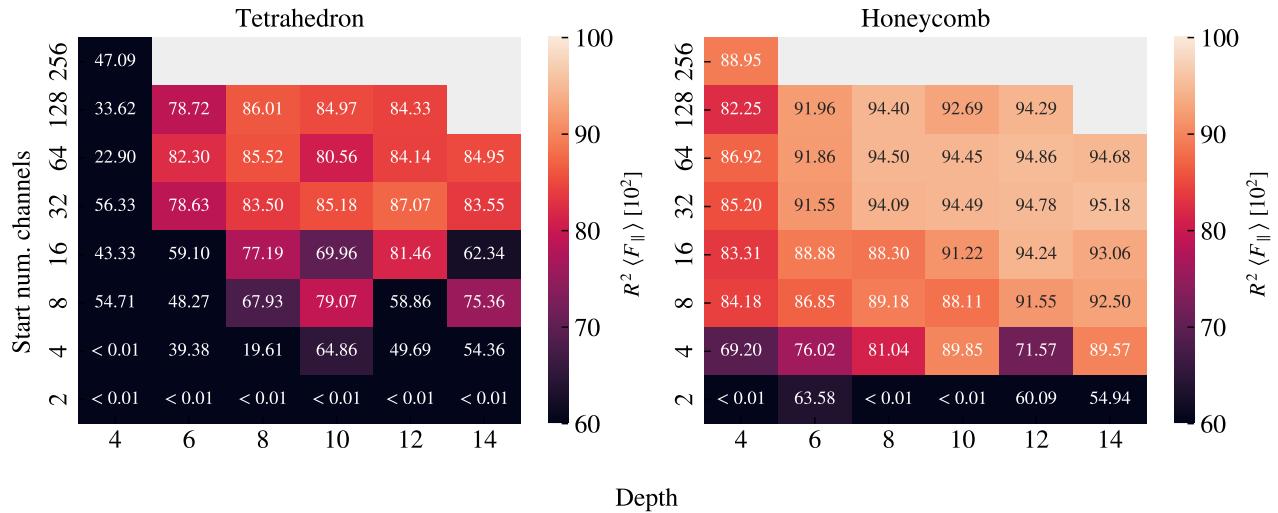


Figure 2.7: Architecture grid search similar to that of Fig. 2.6 on a selected set of configurations. The selected set consists of the top 10 candidates for the max drop property for the Tetrahedron and Honeycomb category in the dataset respectively.

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 looking at the best epoch, we find that models of low depth result in a later best epoch 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, but 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 set in Fig. 2.7 we find significantly lower R^2 scores, especially for the Tetrahedron pattern. Considering, that some of these data points are also present in the training data, this shows clearly that these configurations are more challenging to predict. 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 as it is among the top candidates for the R^2 score and loss in Fig. 2.6 and the R^2 score for the Honeycomb pattern in Fig. 2.7 as well. Hence, we settle on this architecture.

Next, we consider momentum m and weight decay λ in the range $m \in [0.85, 0.99]$ and $\lambda \in [0, 1 \times 10^{-2}]$. For each choice of momentum, we perform a learning rate range test. We propose two learning rate schemes: A constant learning rate as used until this point and a one-cycle policy. In the one-cycle policy, we set a maximum bound for the learning rate and start from a factor 1/20 of this bound and increase towards the maximum bound during the first 30% of the training. For the final 70% of training, we decrease towards a final minimum given as a factor 10^{-4} of the maximum bound. 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 loss curve while the maximum bound used for the one-cycle policy 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. 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.

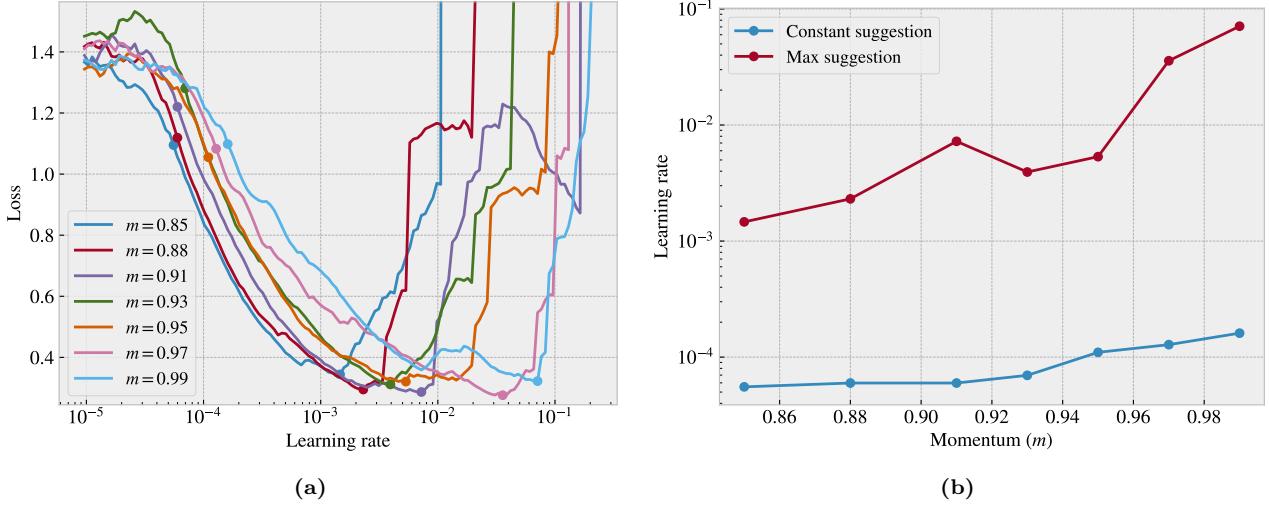


Figure 2.8: Learning rate range test for different momentum values m . (a) show the validation loss as the learning rate is increased exponentially from 10^{-7} towards 10 with increments for each batch iteration during 1 epoch (yielding a total of 242 possible increments). As the curve diverges the 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 for the cyclic learning rate scheme. (b) Shows the corresponding learning rate suggestion for the constant learning rate scheme and the maximum for the cyclic scheme as a function of momentum choice.

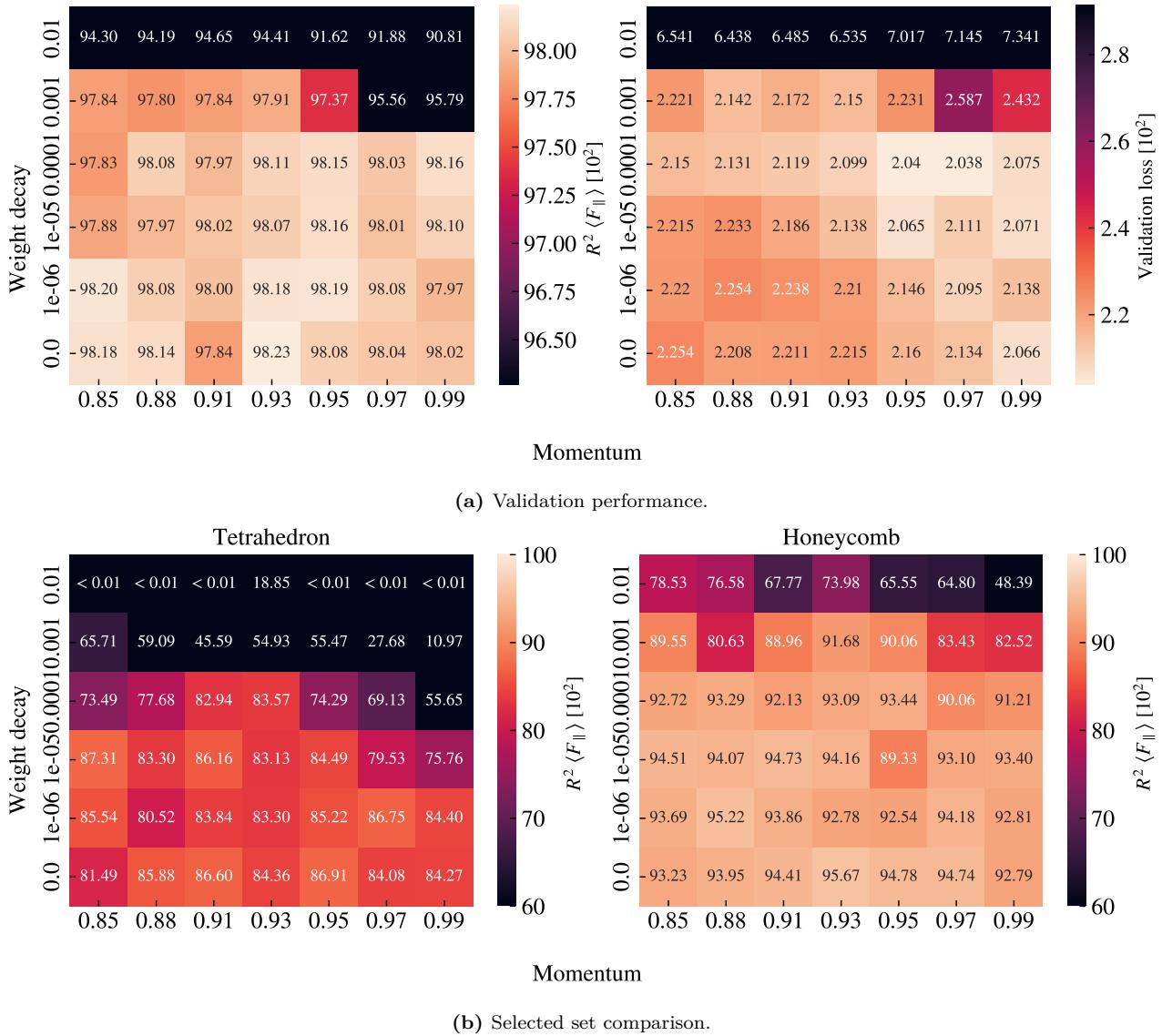


Figure 2.9: Momentum and weight decay grid search using a constant learning rate corresponding to the results from the learning rate range test in Fig. 2.8. (a) Shows the friction mean R^2 validation score (left) and the validation loss (right). (b) Shows the friction mean R^2 validation score for the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

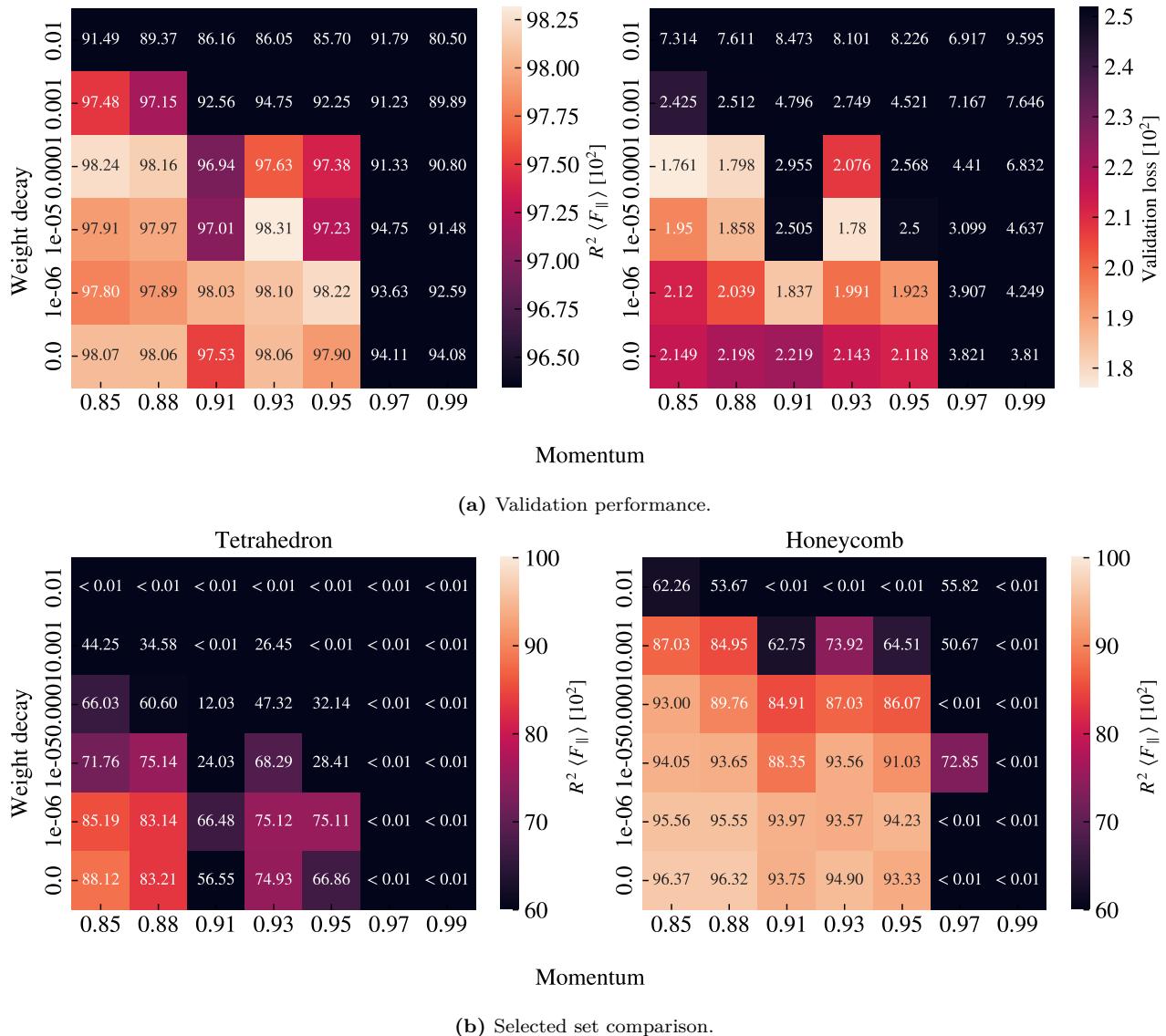


Figure 2.10: Momentum and weight decay grid search using a cyclic learning rate and cyclic momentum scheme. The learning rate maximum bound is chosen according to the learning rate range test in Fig. 2.8. The learning rate starts from a factor 1/20 of the maximum bound, increases towards the maximum 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) Shows the friction mean R^2 validation score (left) and the validation loss (right). (b) Shows 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.

In the one-cycle policy, we set a maximum bound for the learning rate and start from a factor 1/20 of this bound and increase towards the maximum bound during the first 30% of the training. For the final 70% of training, we decrease towards a final minimum given as a factor 10^{-4} of the maximum bound. 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 loss curve while the maximum bound used for the one-cycle policy is

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 can improve these scores 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). However, these values are not taken from the same hyperparameter settings and we did not find a better option across all evaluation metrics. 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 towards a low momentum $m \leq 0.93$ and 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 provided 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 chances of discovering novel designs than to reduce the chance of getting false positive results. Since we retain the option to verify the properties of a given design through MD simulations afterward, we do not have to rely on the machine learning prediction as a final score. Thus we choose the cyclic trained model with low 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, and it would have been natural to perform an extended search in that range. This was omitted due to time prioritization and the belief that the potential gain of doing so was not huge.

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

2.4.5 Final model

From the hypertuning process, we choose the S32D12 model trained by a cyclic training scheme with a momentum of 0.85, an accordingly chosen learning rate from the learning range test 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 the accuracy score of 88% corresponding to never predicting a rupture. Looking at the relative error for the rupture strain we find a large error of $\sim 13\%$. This is lower for the Tetrahedron (5.9%) and Honeycomb (1.5%) set. Thus we might get the idea that the relative error is especially high on the validation set due to a few cases of very low rupture strains in the Random walk patterns which would shift up the average.

Fig. 2.11 shows the mean friction, max friction and relative contact in comparison to the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) used in the pilot study. We note that these configurations are also partly contained within the training data, but this serves as a visualization of how the fit looks for R^2 scores above 98%. We will evaluate a true test set later on based on the proposals from the accelerated search.

Table 2.4: Evaluation of the final model performance considering all the model outputs. Mean values are taken over the score for each configuration in 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

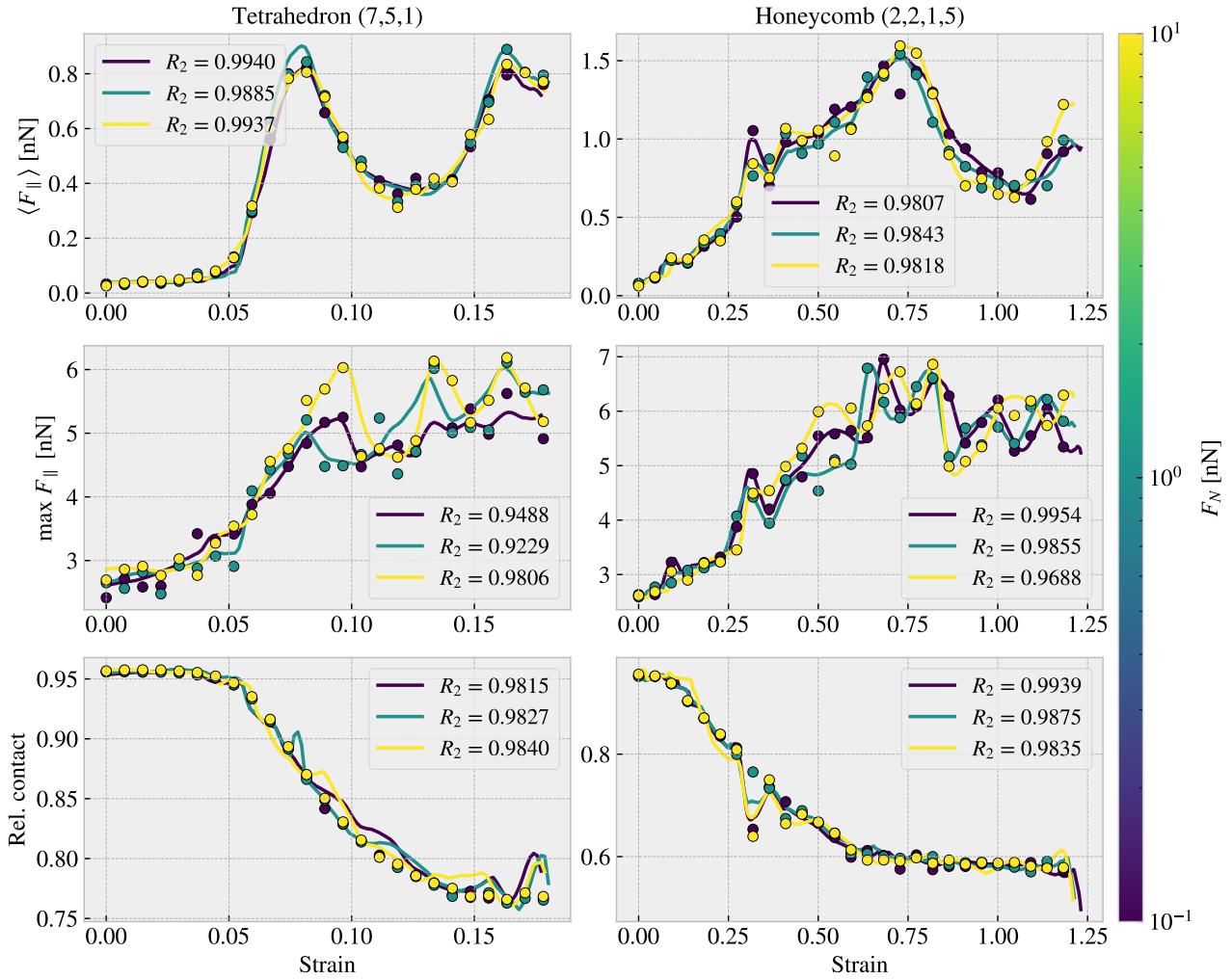


Figure 2.11: Visual evaluation of the final model performance on the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) used in the pilot study, for the mean friction $\langle F_{\parallel} \rangle$, maximum friction $\max F_{\parallel}$ and the relative contact as a function of strain. The model predictions (solid line) is 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) with the color denoting the normal load which. The R^2 scores are shown for each prediction fit for each load value.

Using 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 accordingly. This is shown in Tables 2.5 to 2.7 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 by 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 greater than for the remaining properties. This is supported by the fact that the model predicts the Random walk 12 (Table 2.7) to have negative mean friction, showing a lack of precision. For the maximum categories, we find that the model gives a slightly better ranking for the Tetrahedon 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 in the top 5. This gives an indication that our model can be used to perform an accelerated search of new configurations yielding a meaningful ranking of property scores.

Table 2.5: Tetrahedon

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

Table 2.6: Honeycomb

ML Rank	Data		ML		Data Rank
	Config	Value [nN]	Config	Value [nN]	
min F_{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_{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 ΔF_{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 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

Table 2.7: RW

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

2.5 Accelerated Search

From Sec. 2.4 we have found promising results that we can use a 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 optimization metrics. 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 increment of the parameters will lead to an increased spacing within the pattern. This 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 135×10^3 configurations for the Tetrahedron pattern and 2.025×10^6 for the Honeycomb pattern. For the Random walk generator, we do a Monte Carlo sampling. 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 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: True* and *Centering: False* throughout the search. For the handling of clustering, we implement the repair algorithm such that the sheet is repaired by the least modifications approach rather than retrying the generation several times *Make sure that this is introduced somewhere*. Due to the extra computation time associated with the random walk and the repair algorithm, we only generate 10k configurations within this class. For the ML evaluation of the configurations we use a normal load of 5 nN and generate a strain curve in the domain [0, 2] using 100 evenly spaced points. We compute the properties of interest and rank the configurations accordingly. The top candidates for each property are shown in Table 2.8 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 2.8: Pattern search. The values are 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

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 highlight the weaknesses of the model as we reward an unphysical negative value. In order to resolve this problem one might need 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 2.8 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.

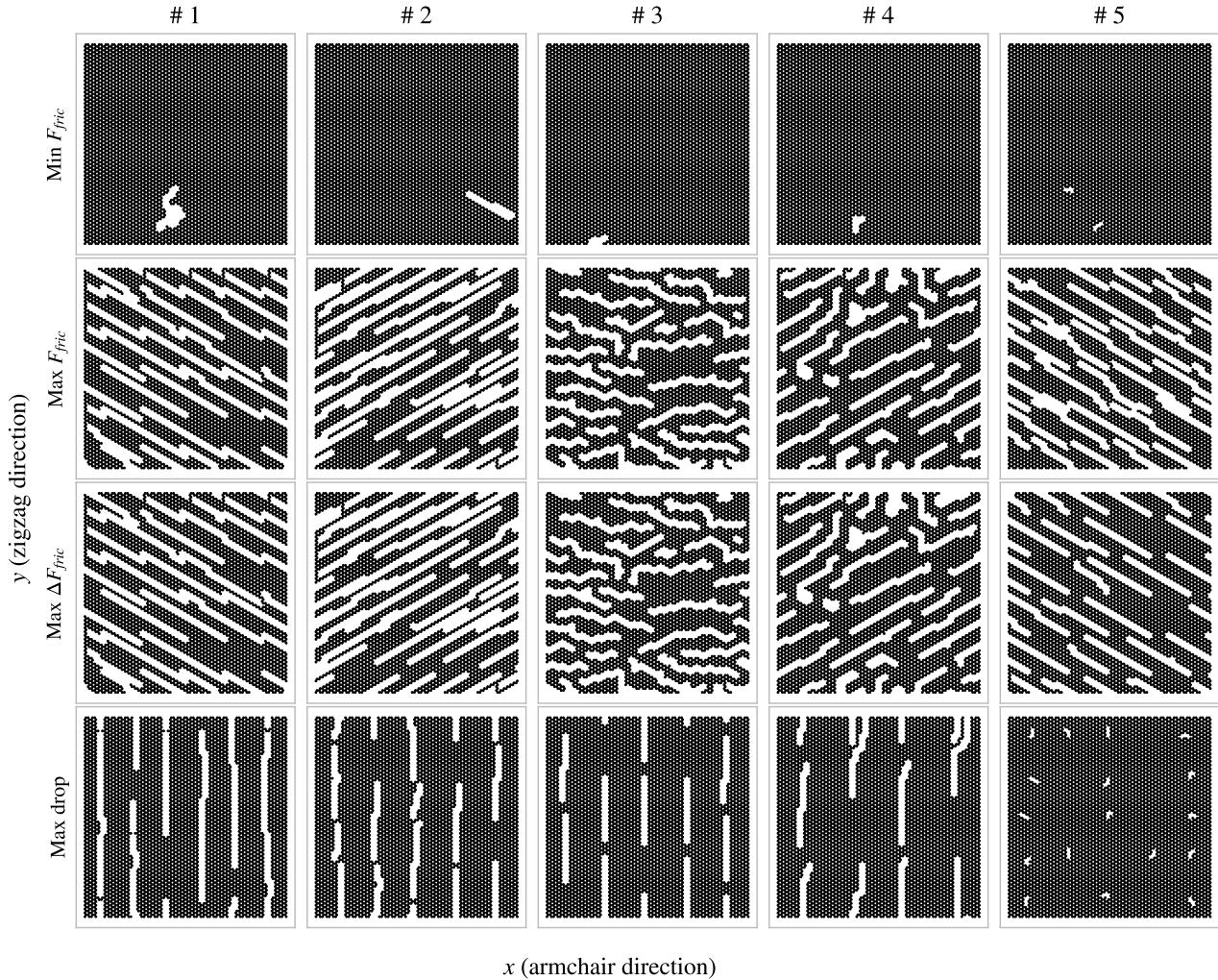


Figure 2.12: RW search top results.

Among the remaining maximum properties, we find competing values for the Honeycomb and Random walk classes. However, the top-scoring values for the Honeycomb correspond to configurations within the original dataset which is also the case for the Tetrahedron top candidates. The only difference is the randomized reference position. When taking a closer look at the 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 scores 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 ML. 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 [refer to theory or directly to the sources?](#), 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 generate 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 we also find that the prediction of the rupture point plays an important role for the max drop property. As 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.

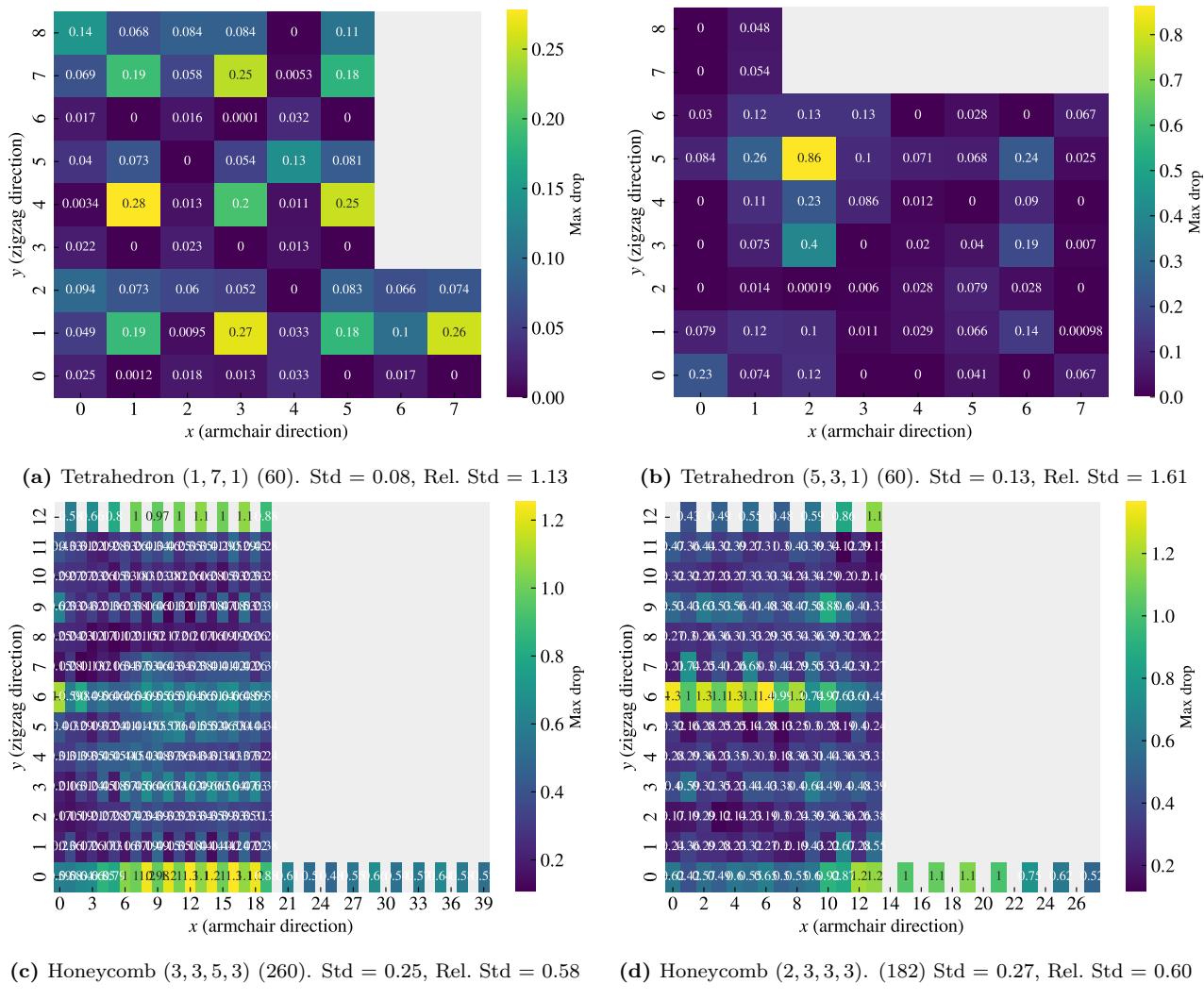


Figure 2.13: CAPTION Fix overlapping annotation text

In order to get 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 candidates for each of the properties of interest in the Random walk search shown in Fig. 2.12. We calculate the ground truth using MD simulation by sampling 30 pseudo uniform strain values using a normal load of 5 nN. Unfortunately, the test set reveals a significantly 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 mean friction average R^2 score is negative which indicates that our model performs worse than simply guessing on a constant value given by 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 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. However, by going back to the hypertuning process and choosing the best model (C16D8) from the architecture complexity grid test given by

the lowest validation score, trained with a constant learning rate, we get similar results. This indicates that the poor test result is not simply caused by the hypertuning choices. Instead, it points to the fact that our original training data does not contain a generalized distribution that accurately captures 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

Despite the fact that 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 studying. 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 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 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 configuration 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.

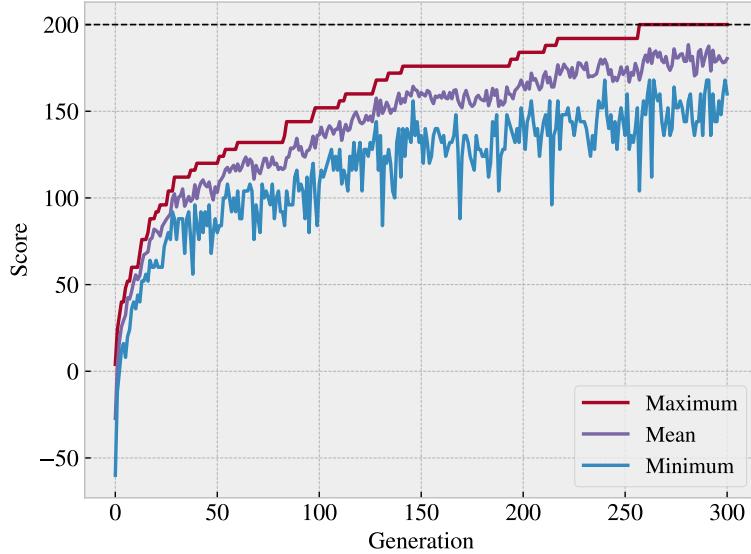


Figure 2.14: Optimization for the maximum Ising energy given by Eq. (2.3) for a 10×10 lattice using a population size of 10. The population was initialized with random noise having a probability of 0.5 being -1 and 0.5 being $+1$. The three curves indicate the minimum, mean, and maximum scores in the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

Moving on from the verification of the algorithm, we consider the optimization for the max drop candidate for the Kirigami patterns. We utilize the machine learning model to evaluate the friction-strain curve for each pattern and compute the corresponding maximum drop property value, similar to what we have done in ???. For the initialization of the population, we take a basis in the top candidates for the pattern generation search from ???. 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 max candidates within the Tetrahedron, Honeycomb and Random walk 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 score individual at generation 0 did not improve throughout the search, even though the average score was rising initially. For the Random walk, we perform 5 runs for different population initializations. For each population, we use the parameters corresponding to the top 5 max drop candidates from the Random walk pattern generation search (seen in the bottom row in Fig. 2.12) respectively. However, 4 out of 5 runs gave a similar result as seen for the Tetrahedron and Honeycomb patterns, meaning that only a single run provided a new pattern for the top 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 porosity of 0.25 and 0.5 respectively. This time the algorithm improved the top 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 mainly attribute this to our finding that the machine learning model may not be 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 top candidate, the Grad-CAM method highlights some areas in the noise configuration as contributing more positively than others, but we do not find any obvious trends 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” towards 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, that these results should be taken carefully due to the instability of the model, we note this as a topic for further investigations.

One possible explanation is that the dissipation of energy related to the thermostat in the pull blocks might be of importance. Given the instability of the model, we caution that these results should be interpreted with care, but we consider this as a potential topic for further investigation.

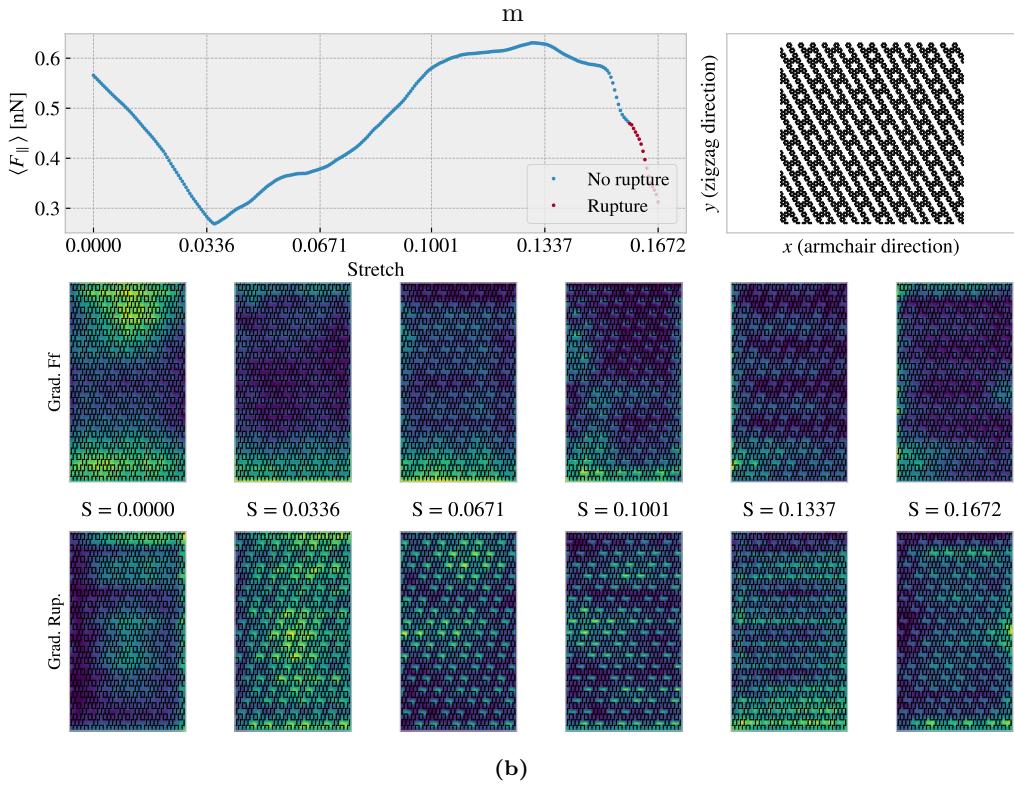
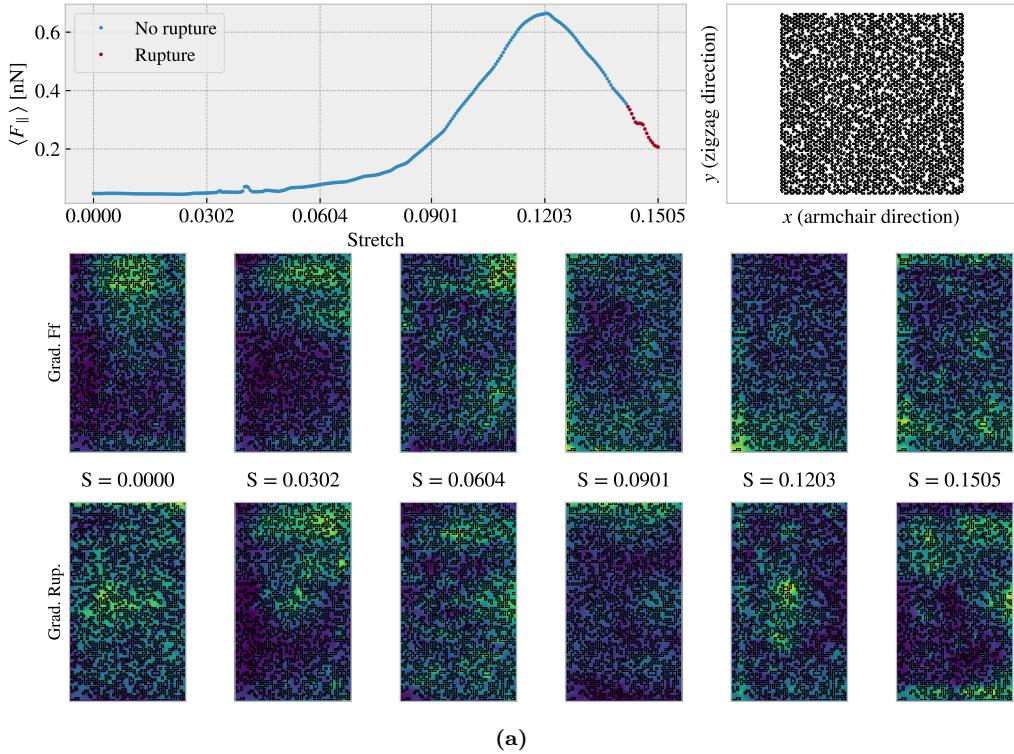


Figure 2.15: (The figure continues on the next page)

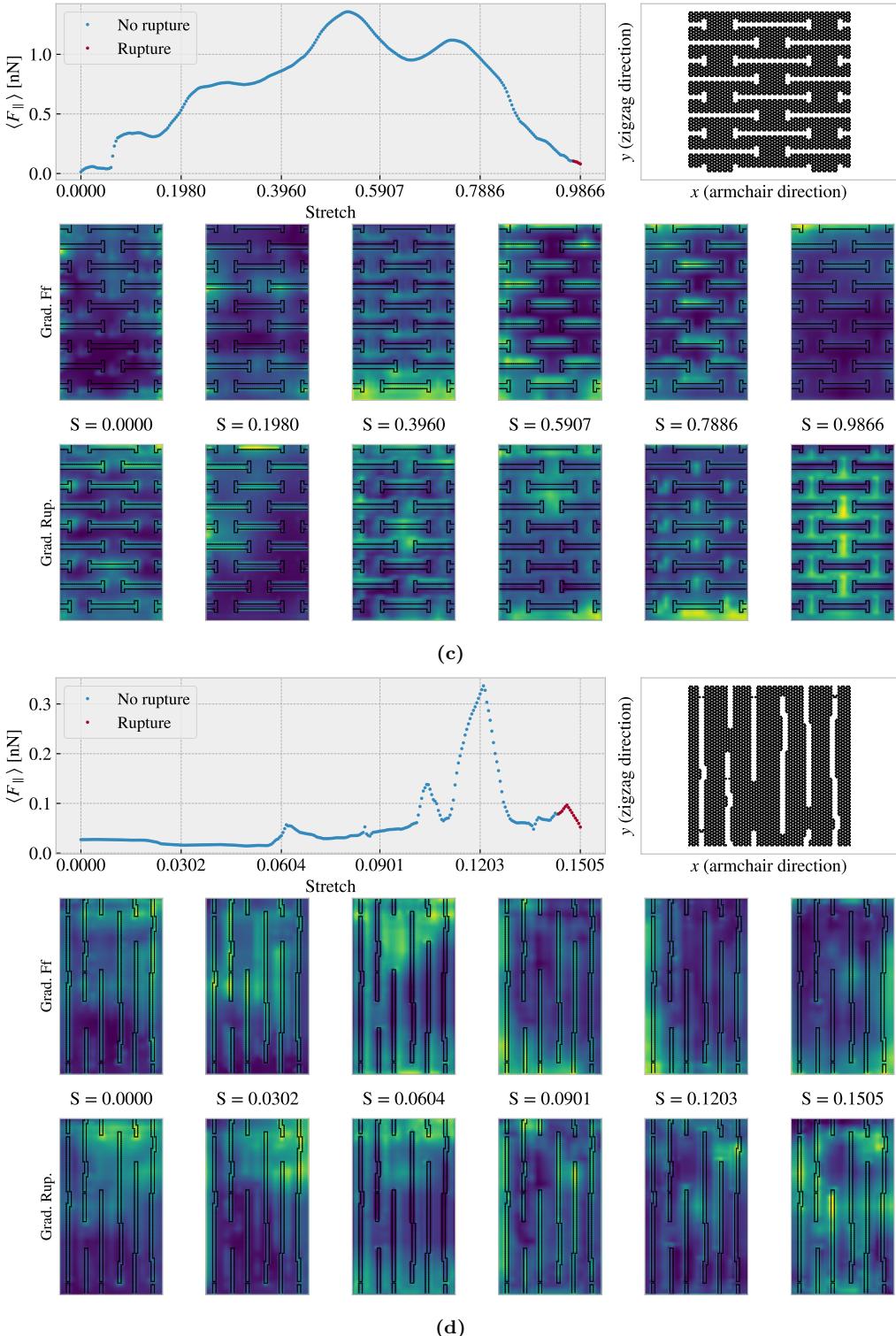


Figure 2.15: Grad-CAM analysis of selected Kirigami configurations. The top rows show the predicted friction-strain curve (left) with the rupture prediction indicated by the colors (blue: no rupture, red: rupture), and a visualization of the Kirigami grapheme sheet on the hexagonal lattice (right). The remaining rows show the grad-CAM heatmaps with respect to the prediction of the friction (Grad. Ff) 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 as shown in Fig. 2.12.

Chapter 3

Summary

In this thesis, we have studied the nanoscale friction of a Kirigami graphene sheet under the influence of load and strain using MD simulations. We have developed a numerical framework for generating diverse Kirigami designs which we have utilized to create a dataset for the frictional behavior depending on Kirigami pattern, strain, and loading. Our findings suggest that the frictional behavior of a Kirigami sheet is highly dependent on the geometry of the pattern and the strain conditions. We observed that the out-of-plane buckling can be associated with a non-linear friction-strain curve which can be utilized to demonstrate a negative friction coefficient in a system with coupled load and strain. Moreover, we have investigated the possibility to use machine learning on this dataset and attempted an accelerated search. Our findings imply that machine learning can be feasible for this approach, additional data is required to establish a more reliable foundation for a search for new Kirigami patterns. In this chapter, we will provide a summary of our findings and draw conclusions based on the results obtained. Finally, we will suggest some topics for further research.

3.1 Summary and conclusions

3.1.1 Designing an MD simulation

We have designed an MD simulation for the examination of friction for a graphene sheet sliding on a silicon substrate. The key system features were the introduction of the pull blocks, defined as the end regions of the sheet with respect to the sliding direction, which was utilized for applying normal load and sliding the sheet. The pull blocks were made partly rigid and used to employ a thermostat as well. Through an analysis of the friction forces retrieved from sliding simulations, we have established a standardized metric for kinetic friction. In particular, we measured the force exerted by the substrate on the full sheet, including the pull blocks, with respect to the sliding direction, and determined the kinetic friction as the mean value of the last half of the simulation. The uncertainties were estimated based on the fluctuations in the running mean. We found that the assessment of static friction was ambiguous for our simulation and did not pursue this further. From the analysis of the force traces, friction force vs. time, we identify the friction behavior in our simulation domain as being in the smooth sliding regime. This is attributed to the choice of a relatively high sliding speed (20 m/s) and infinitely stiff springs for the tethering of the sheet. This was supported by a demonstration of a transition to the stick-slip regime through the use of softer springs and a decrease in sliding speed. By conducting a more systematic investigation of the effects of temperature, sliding speed, spring constant and timestep, we identified a set of default values based on numerical stability and computational cost. During this process, we aimed to select the variables that would maintain a relatively stable measured friction with moderate perturbations around these default values. We found that friction increased with temperature which we attribute to being in the ballistic sliding regime. In the absence of clear indications from the investigation regarding an appropriate temperature, we opted for the standard choice of room temperature, 300 K. Furthermore, we found friction to increase with velocity as expected, with some signs of phonon resonance at certain sliding speeds as well. We chose a rather high velocity of 20 m/s mainly for the consideration of computational costs. For the spring constant, we found decreasing friction with increasing stiffness of the springs. This is associated with the transition from a stick-slip-influenced regime toward smooth sliding and can be attributed to an underlying change in commensurability. The choice of an infinitely stiff spring was made from an assessment of the variation

in friction with perturbations in the spring constant value. Finally, we confirmed that a timestep of 1 fs provides reasonable numerical stability. However, based on fluctuations with timestep we find that the uncertainty in the simulations might be higher than first estimated. For the non-strained Kirigami sheet, these fluctuations were on the order of ± 0.017 nN for the evaluation of the kinetic friction.

3.1.2 Generetig Kirigami patterns

In order to investigate the effects of Kirigami design we have created a numerical framework for generating various patterns. By defining an indexing system for the hexagonal lattice structure we were able to define the Kirigami designs as a 2D binary matrix for numerical implementation. We have selected two macroscale designs, which we denote the *Tetrahedron* and *Honeycomb* patterns, based on their ability to exhibit out-of-plane buckling when subjected to strain. By digitalizing the designs to match the hexagonal graphene lattice, we found that the characteristic design features can be translated to the nanoscale, as we observed similar out-of-plane buckling in MD simulations. Through our numerical framework we were able to create an ensemble of perturbed unique variations which yielded approximately 135×10^3 and 2.025×10^6 unique configurations for the Tetrahedron and Honeycomb patterns respectively. When considering the possibility to translate the periodic patterns on the graphene sheet the number of possible patterns can be increased by approximately a factor of 100. To introduce some random design features, we have developed a framework for generating Kirigami patterns based on random walks. The framework includes mechanisms such as bias, avoidance of existing cuts, preference for maintaining a direction, and procedures for repairing the sheet for simulation purposes. In general, we found that the capabilities of this numerical framework for generating Kirigami designs exceeded our computational resources with regard to performing MD simulation under different loads and strains for each of the designs. In general, we found that the computational capacity of our numerical framework for generating Kirigami designs surpassed our available resources for conducting MD simulations under various load and strain for each design. Our MD-based dataset only utilized a subset of configurations with 9660 data points based on 216 Kirigami configurations (Tetrahedron: 68, Honeycomb: 45, Random walk: 100, Pilot study: 3). Hence, we argue that the Kirigami generative framework can be valuable for further studies on an extended dataset.

3.1.3 Control friction using Kirigami

We have investigated the frictional behavior of the Tetrahedron and Honeycomb patterns in comparison to a non-cut sheet under various strains and loads. Initially, we observed that straining the Kirigami sheets in a vacuum resulted in an out-of-plane buckling. When adding the substrate to the simulation this translated into a decreasing contact area with strain. We found the Honeycomb sheet to exhibit the most significant buckling with a corresponding reduction of relative contact to approximately 43%, whereas the non-cut sheet did not produce any significant buckling in comparison. For the Kirigami sheets, we found that friction initially increased with strain, which made for increasing friction with decreasing contact area. As the strain continued to increase the friction-strain curve exhibited highly non-linear trends with strong negative slopes as well (see ??). During the full strain, the contact area was decreasing monotonically except for a slight increase just before rupturing. These results contradict the asperity theory hypothesis of decreasing friction with decreasing contact area, which is also supported by the predictions of the 2D Frenkel-Kontorova models suggesting increasing friction with (contacting) system size. Thus, we conclude that the contact area is not a governing mechanism for the friction-strain relationship observed. From the investigation of the friction-load relationship, we found that both friction and contact area increased slightly with load, but we were not able to find significant evidence of any linear relationship between friction and contact area as predicted by asperity theory. The non-cut sheet did not show any dependency on the strain as both the friction and the contact area remained constant with strain. Thus our findings suggest that a changing contact area and the strain-induced friction effects might be associated with an underlying mechanism connected to the buckling of the sheet. We did find a small independent effect from the introduction of Kirigami cuts before applying strain. We observed a small increase in friction between the non-cut sheet and the Kirigami sheet, but this was one order of magnitude lower than the effects associated with the strain in combination. On the other hand, the non-cut sheet did not exhibit any significant effects from strain and we conclude that the independent effects from Kirigami or strain-induced tension cannot be regarded as a dominant mechanism on its own. When considering the friction dependency with load, we generally found a weak dependency corresponding to a friction coefficient on the order of 10^{-4} – 10^{-5} even though we could not confirm any clear relationship. This is best attributed to a superlubric state of the graphene sheet as seen in

other studies as well. The slope of the friction-load curve was not considerably affected by the straining of the Kirigami sheet which led us to the conclusion that strain-induced effects are dominant in comparison to any load-related effects.

When considering our findings in light of previous related results we find it plausible that the governing mechanism for the observed friction effects is related to commensurability as predicted by the Frenkel-Kontorova models. Since we have observed extremely low friction in our system, we argue that the non-deformed sheet corresponds to an incommensurable configuration. This is supported by the fact that we were not able to lower the friction from the original starting point. Additionally, this also aligns with the observation that the introduction of the non-strained Kirigami designs increased friction slightly. Since some atoms were removed from the lattice this can be hypothesized to relax the incommensurability to some degree. When the Kirigami sheet buckles during stretching, it allows for a considerable rearrangement of the atoms in contact with the substrate. Hence, it may transition in and out of commensurable configurations, which could explain the non-linear trend for the friction-strain curve. One way to test this hypothesis is to reorientate the sheet in order to reach a commensurable start configuration. Then we might find a transition from a commensurable to an incommensurable case during the initial straining which would result in a lowering in friction with respect to the starting point.

3.1.4 Capturing trends in friction data with machine learning

By utilizing the numerical framework for generating Kirigami design we have created a MD-based dataset for the frictional behavior depending on Kirigami designs, load and strain. The dataset reveals some general correlations with mean friction, such as a positive correlation to strain (0.77) and porosity (0.60), and a negative correlation to contact area (-0.67). These results align with the finding in the pilot study, suggesting that the change in friction is associated with cuts in the sheet (porosity) and a changing contact area.

By defining the friction property metrics: $\min F_{\text{fric}}$, $\max F_{\text{fric}}$, $\max \Delta F_{\text{fric}}$ and $\max \text{drop}$ (maximum decrease in friction with strain), we investigated the top design candidates within our dataset. From these results, we found no indication for the possibility to reduce friction with the Kirigami approach since the non-cut sheet provided the lowest overall friction. Furthermore, among the top candidates, we found that a flat friction-strain profile is mainly associated with little decrease in the contact area and vice versa. These observations are consistent with the results of the pilot study and support the hypothesis that commensurability plays a key role in governing the behavior of the system. In terms of the maximum properties, we observed an improvement compared to the values obtained in the pilot study, with the Honeycomb pattern exhibiting the highest scores. This indicates that the data may be useful for optimizing these properties.

For the machine learning investigation, we have implemented a VGGNet-16-inspired convolutional neural network with a deep “stairlike” architecture: C32-C64-C128-C256-C512-C1024-D1024-D512-D256-D128-D64-D32, for convolutional layers C with the number denoting channels and fully connected (dense) layers D with the number denoting nodes. The final model contains 1.3×10^7 parameters and was trained using the ADAM optimizer for a cyclic learning rate and momentum scheme. We trained the network for a total 1000 epochs while saving the best model during training based on the validation score. The model validation performance gives a mean friction R^2 score of $\sim 98\%$ and a rupture accuracy of $\sim 96\%$. However, we got lower scores for a selected subset of the Tetrahedon ($R^2 \sim 88.7\%$) and Honeycomb ($R^2 \sim 96.6$) pattern based on the top 10 max drop scores respectively. The scores obtained were lower, even though the selected configurations were partly included in the training data and the hyperparameter selection favored the performance on this selected set. Thus we conclude that these selected configurations, associated with a highly non-linear friction-strain curve, represent a bigger challenge for the machine learning prediction. One interpretation is that these involve the most complex dynamics and perhaps that this is not readily distinguished from the behavior of the other configurations which constitute the majority of the data set. By evaluating the ability of the model to rank the dataset based on property scores, we found that it was able to effectively represent the top three scores for the maximum categories. However, the ranking for the minimum friction property was lacking. We attribute this discrepancy to the fact that this property requires a higher level of precision, which the model was not able to achieve. To obtain a more accurate evaluation of the model’s performance, we generated a test set using MD simulations for the random walk based suggestions obtained from the accelerated search. The results showed a significantly worse performance compared to the validation set, with a two-order of magnitude higher loss and a negative R^2 score for the mean friction property. The negative R^2 score suggests that the model’s predictions were worse than simply predicting the mean value of the true data.

However, by reevaluating the hypertuning and choosing solely on the basis of validation loss, we still found poor results on the test set. This suggests that the inadequate performance is not solely due to biased hypertuning but rather because our original dataset did not sufficiently cover a diverse range of Kirigami configurations to accurately capture the full complexity of its frictional behavior. The validation scores indicate that the use of machine learning is a feasible approach for addressing this problem, as we were able to identify some general trends in the data. Nonetheless, from the test scores, it is evident that further improvements to the dataset are necessary in order to develop a reliable model.

3.1.5 Accelerated search

Using the ML model we performed two types of accelerated search. One by evaluating the property scores of an extended dataset and another with the use of the genetic algorithm approach. For the extended dataset search, we used the developed pattern generators to generate 1.35×10^6 Tetrahedron, 2.025×10^7 Honeycomb and 10 k Random walk patterns. The search results for the minimum friction property indicate a preference for low cut density or low porosity. This aligns with the overall observation that the dataset does not provide any suggestions to further reduction in friction, as the non-cut sheet has the lowest friction.

The search for the maximum properties resulted in some minor score increases but the suggested candidates were mainly overlapping with the original dataset. By investigating the sensitivity to translations of the Tetrahedron and Honeycomb patterns, we observed significant variations in the model's predictions with only minor translations. This can be attributed to a physical dependency since these translations affect the edge of the sheet. However, given the poor model performance on the test set, we believe it is more likely that this variation is due to an insufficiency in the model caused by the limitations of the dataset.

In our investigation of the genetic algorithm approach, we used a starting population that was based on the results from the extended dataset accelerated search, as well as some randomly generated initializations with different porosity values. However, this approach did not provide any noteworthy indication for new design structures worth more investigation. In general, the initialization of the population itself proved to be a more promising strategy than the genetic algorithm. We acknowledge that further effort could potentially yield useful results with the genetic algorithm approach. However, we believe that the current lack of promising results can be attributed to the uncertainty of the model which where the reason for not pursuing this any further.

By considering the Grad-CAM explanation method, we observed that the model predictions were often substantially reliant on the top and bottom edges of the Kirigami configurations. This was unexpected since these edges are not true edges but are connected to the pull blocks used in the simulation. Despite the model's uncertain predictions, we speculate that this may be due to the thermostat effects from the pull blocks. Therefore, we note this as a feature worth investigating in the simulations.

3.1.6 Negative friction coefficient

Based on our initial investigations of the Kirigami sheet, we have discovered a highly non-linear friction-strain relationship. By proposing a linear coupling between load and strain with ratio R , we find that these results suggest the possibility to utilize the negative slope on the friction-strain curve to achieve a negative friction coefficient. Based on the drop from top to bottom of these curves, using the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) pattern from the pilot study, we estimate that the average coefficient within this range will be $-R12.75 \text{ nN}$ for the Tetrahedron pattern and $-R \cdot 2.72 \text{ nN}$ for the Honeycomb pattern.

To investigate this hypothesis, we conducted a simulation with a coupling between load and sheet tension, mimicking a nanomachine attached to the sheet, using the Kirigami configurations from the Pilot study. We observed that the non-linear behavior in the friction-strain curve also translated into a non-linear friction-load relationship for the coupled system. Additionally, we found that the Honeycomb pattern exhibited a non-linear strain-tension curve which resulted in an almost discontinuous increase in friction for the initial increase in load. We attribute this feature to an unfolding process visually confirmed from the simulation frames. For the coupled system with a load-to-tension ratio of 6 we found regions in the friction-load curve with significant negative slopes. By considering the maximum and minimum points for such regions we estimated the average friction coefficient to be -0.31 in the load range $F_N = [4.65, 6.55] \text{ nN}$ for the Tetrahedron pattern and -0.38 in the range $F_N = [0.71, 4.31] \text{ nN}$ for the Honeycomb pattern. These results can be scaled by adjusting the load-to-tension ratio.

Based on our investigations, we have found that the combination of Kirigami cuts and strain has significant potential for controlling friction. Specifically, we have demonstrated that by enforcing a coupling between load and strain through tension, it is possible to achieve a negative friction coefficient. Therefore, we believe that this approach could be promising for developing novel friction-control strategies.

3.2 Outlook

In this thesis, we have demonstrated that certain Kirigami designs exhibit non-linear friction behavior with strain. This discovery was made through an exploration of different designs, which invites further investigation into the underlying mechanisms of this phenomenon. To this end, it would be valuable to one or two selected designs, such as the Tetrahedron and Honeycomb patterns, and study the effects of various physical parameters on the friction-strain curve.

First of all, we suggest an investigation of how the friction-strain curve depends on temperature, sliding speed, spring constant, and loads for an increased range $F_N > 100nN$. This is especially interesting in the context of physical conditions leading to a stick-slip behavior since our study takes a basis in smooth sliding friction. Moreover, it would be valuable to verify that the choices for relaxation time, pauses, interatomic potentials and substrate material are not critical for the qualitative observations. Especially the Adaptive Intermolecular Reactive Empirical Bond Order (AIREBO) potential for the modeling of the graphene sheet might be of interest. In this context, it might also be useful to investigate the effects of excluding adhesion from the simulations.

In order to investigate the hypothesis of commensurability as a governing mechanism we suggest an investigation of the friction-strain curve for different scan angles. If commensurability is an important factor, we hypothesize that the friction-strain curve will exhibit different qualitative shapes at varying scan angles. Additionally, it may be interesting to investigate the friction-strain relationship under a uniform load to gain insight into how the loading distribution affects the out-of-plane buckling and associated commensurability effect.

Another topic worth studying is the relation to scale and edge effects. This includes an investigation of scaling effects, considering the ratio of the sheet area to the edge length, but also a translation of the sheet patterns to study the presence of any Kirigami-induced edge effects. The latter is motivated by the findings from the machine learning predictions. With this regard, we would also suggest a more detailed study of the effect of the thermostat in the pull blocks which is suggested to have a possible importance when judging from the Grad-CAM analysis.

Regarding the machine learning approach, our findings indicate that there is a significant need to expand the dataset. In order to get more insight into this issue one could use unsupervised clustering techniques like the t-Distributed Stochastic Neighbor Embedding (t-SNE) to visualize the distribution of Kirigami configurations in the dataset. Another valuable approach is the active learning method similar to that used by Hanakata et al. [6]. That is, we extend the dataset using the top candidates of a machine learning-driven accelerated search and repeat the process of training the model, searching for new candidates and extending the dataset. This provides a direction for the extension of the dataset which could lead to a more efficient approach to address the complex space of Kirigami designs. We note that one can also create a dataset of a fixed kirigami designs and vary the physical parameters to support the investigations mentioned above. For both variations we believe that the results could benefit from a consideration of more advanced model architectures and machine learning techniques. If successful this would invite further studies of inverse design methods such as GAN or diffusion models.

Appendices

Bibliography

- ¹E. Gnecco and E. Meyer, *Elements of friction theory and nanotribology* (Cambridge University Press, 2015).
- ²Bhusnur, “Introduction”, in *Introduction to tribology* (John Wiley & Sons, Ltd, 2013) Chap. 1, 1–?
- ³H.-J. Kim and D.-E. Kim, “Nano-scale friction: a review”, *International Journal of Precision Engineering and Manufacturing* **10**, 141–151 (2009).
- ⁴K. Holmberg and A. Erdemir, “Influence of tribology on global energy consumption, costs and emissions”, *Friction* **5**, 263–284 (2017).
- ⁵B. 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.
- ⁶P. 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).
- ⁷P. 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).
- ⁸L.-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).
- ⁹Y. Mao, Q. He, and X. Zhao, “Designing complex architectured materials with generative adversarial networks”, *Science Advances* **6**, eaaz4169 (2020).
- ¹⁰Z. 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).
- ¹¹A. 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).
- ¹²S. 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).
- ¹³OpenAI, *Dall-e2*, 2023.
- ¹⁴Midjourney, *Midjourney*, 2023.
- ¹⁵Z. 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).
- ¹⁶B. 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).
- ¹⁷D. Mandelli, W. Ouyang, O. Hod, and M. Urbakh, “Negative friction coefficients in superlubric graphite–hexagonal boron nitride heterojunctions”, *Phys. Rev. Lett.* **122**, 076102 (2019).
- ¹⁸R. W. Liefferink, B. Weber, C. Coulais, and D. Bonn, “Geometric control of sliding friction”, *Extreme Mechanics Letters* **49**, 101475 (2021).
- ¹⁹M. Metzsch, *Tuning Frictional Properties of Kirigami Altered Graphene Sheets using Molecular Dynamics and Machine Learning*.

- ²⁰A. 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).
- ²¹E. M. Nordhagen, *LAMMPS simulator*.
- ²²A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, 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. Kolsbjergr, 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).
- ²³A. 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.