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

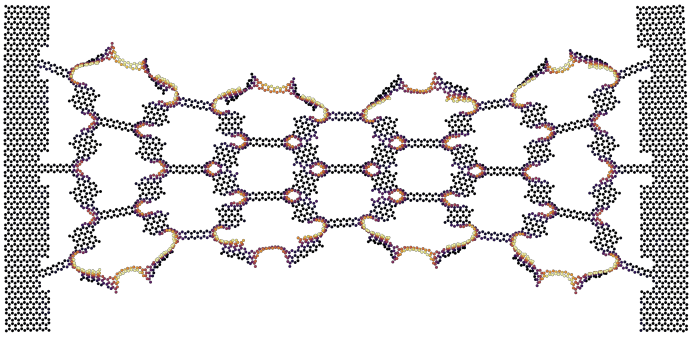
UNIVERSITY OF OSLO

Spring 2023

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

Designs for a negative friction coefficient.

Mikkel Metzsch Jensen



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

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

Printed: Reprosentralen, University of Oslo

Abstract

Abstract.

Acknowledgments

Acknowledgments.

List of Symbols

F_N Normal force (normal load)

Acronyms

CNN Convolutional Neural Network. 15

MD Molecular Dynamics. 1, 2, 3, 9, 15, 28

ML Machine Learning. 2, 3, 24, 25, 26, 27, 28

MSE Mean Squared Error. 17

Contents

Chapter 1

Introduction

1.1 Motivation

Friction is the force that prevents the relative motion of objects in contact. Even though the everyday person might not be familiar with the term *friction* 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 tremendous 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 "togglable" 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 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 make for seemingly intractable 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 with the latter being used in state-of-the-art AI systems such as OpenAI's DALL·E2 or Midjourney SOURCE?. By working with a so-called *encoder-decoder* network structure, one can build a model that reverses the prediction process. That is, the model predicts a design from a set of physical target properties. In the papers by Hanakata et al. both the *accelerated search* and the *inverse design* approach was proven successful to create novel metamaterial kirigami designs with the graphene sheet.

Hanakata et al. attribute the variety in yield 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 rarely found property which is mainly found for the unloading phase of adhesive surfaces [13] or for the loading phase of particular heterojunction materials [14, 15]

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. [16] 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 regime would involve a different set of underlying mechanisms and thus contribute to new insight in this field.

1.2 Goals

In this thesis, we investigate the possibility to alter 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. With the use of machine learning (ML), we perform an accelerated search of designs to explore new designs. The main goals of this thesis can be summarized as follows.

1. Design a robust MD simulation procedure to evaluate the frictional properties of a Kirigami graphene sheet under specified physical conditions.
2. Develop a numerical framework for creating various Kirigami designs, both by seeking inspiration from macroscale designs and by the use of stochastically based algorithms.
3. Investigate the friction behavior under varying load and stretch for different Kirigami designs.
4. Develop and train a ML model to predict the MD simulation result and perform an accelerated search of new designs for the scope of optimizing certain frictional properties.

1.3 Contributions

What did I actually achieve Include Githib link

1.4 Thesis structure

In Part I: Background Theory, we cover the theoretical background related to Friction (**??**), Molecular Dynamics (**??**) and Machine Learning (**??**).

In **??**: Friction, we introduce the most relevant theoretical concepts of friction through a division by scale: Macroscale (**??**), Microscale (**??**) and nanoscale (**??**). We emphasize the nanoscale since this is of the most importance for our study. This is followed by a summary of relevant experimental and numerical results **??** and a more formal specification of our research questions (**??**).

In **??**: Molecular Dynamics, we introduce the main concepts related to the simulations used in this thesis. The main parts involve a description of the potentials used (**??**), the numerical solutions (**??**) and the modeling of temperature (**??**)

In **??**: Machine Learning, we introduce the basics of machine learning through a general presentation of the neural network **??** followed by the convolutional network (**??**) which we will use in our study. Additionally, we discuss a strategy for choosing model hypertuning (**??**) and a simple approach for model prediction explanations (). Finally, we introduce a version of the genetic algorithm applicable for accelerated search based on a machine learning model (**??**).

In Part II: Simulations, we define our numerical procedure and present and discuss the main findings of this thesis.

In **??**: System, we ...
In **??**: Pilot study, we ...
In Chapter 2: XXX ...
In **??**: XXX, ...
The thesis is summarized in **??**

Additional figures are shown in **??**, **??** and **??**. get appendix with only letter A., B. and C.

Part I

Background Theory

Part II

Simulations

Chapter 2

Kirigami configuration exploration

Based on the findings in the Pilot Study **??** we will further explore the effects of Kirigami designs for stretch-depending friction. We are especially interested in the optimization toward friction force and friction coefficient extrema. First, we create a dataset through MD simulations for an extended set of Kirigami designs. This provides useful insight regarding general trends for the frictonal behavior. Following, we use the dataset to investigate the possibility to use machine learning for a prediction of the friction behavior based on Kirigami design, stretch and load. Finally, we attempt an accelerated search using the acquired machine learning model.

2.1 Generating the dataset

We aim to create a dataset that contains information on frictional behavior for varying Kirigami configurations (is it bad to change from "design" to "configuration"...), stretch and load. For each configuration, we sample 15 pseudo uniform stretch values (see **??**) between zero and the rupture stretch according to the rupture test. The normal force is uniformly sampled in the range $[0.1, 10]$ nN. In total, this gives 3×15 data points for each configuration. For the remaining parameters, we use the values presented in the pilot study (see **??**). We are mainly concerned with the mean friction and whether the sheet is ruptured or not. However, we also include the maximum friction, the relative contact, the rupture stretch (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. A summary of the dataset is given in Table 2.1 while all configurations are shown explicitly in **??**. The Tetrahedron and Honeycomb parameters are chosen to provide additional variations of the evaluated configurations evaluated in **??** which exhibited interesting properties, while the Random walk configurations are chosen to get as diverse configurations as possible. Notice that not all submitted data points "make it" to the final dataset. This is due to a small bug in the data generation procedure[1].

[1]The 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 stretch limit, but also with a new random velocity and thermostat initialization. The sheet is then stretched and checkpoints of the simulation state (LAMMPS restart files) are saved for each of the targeted stretch samples. However, if the rupture point arrives earlier than suggested by the rupture test due, to the randomness from the initialization, some of the planned stretch samples might get lost without a corresponding checkpoint 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 quite easily have been mitigated by a rewrite of the code, but it was first discovered after the dataset had been created. We notice, however, that the dataset still contains 11.57 % rupture events which provide a reasonable amount of rupture events to incorporate in the machine learning model

Table 2.1: Summary of the number of generated data points in the dataset. Due to slight deviations in the rupture stretch and the specific numerical procedure not all submitted simulations "make it" to 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

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

where $\mathrm{Cov}(X,Y)$ is the covariance, μ the mean value and σ the standard deviation. The correlation coefficients range from a complete negative correlation (-1) through no correlation (0) to a complete positive correlation (1). The correlation coefficients is shown in Fig. 2.1

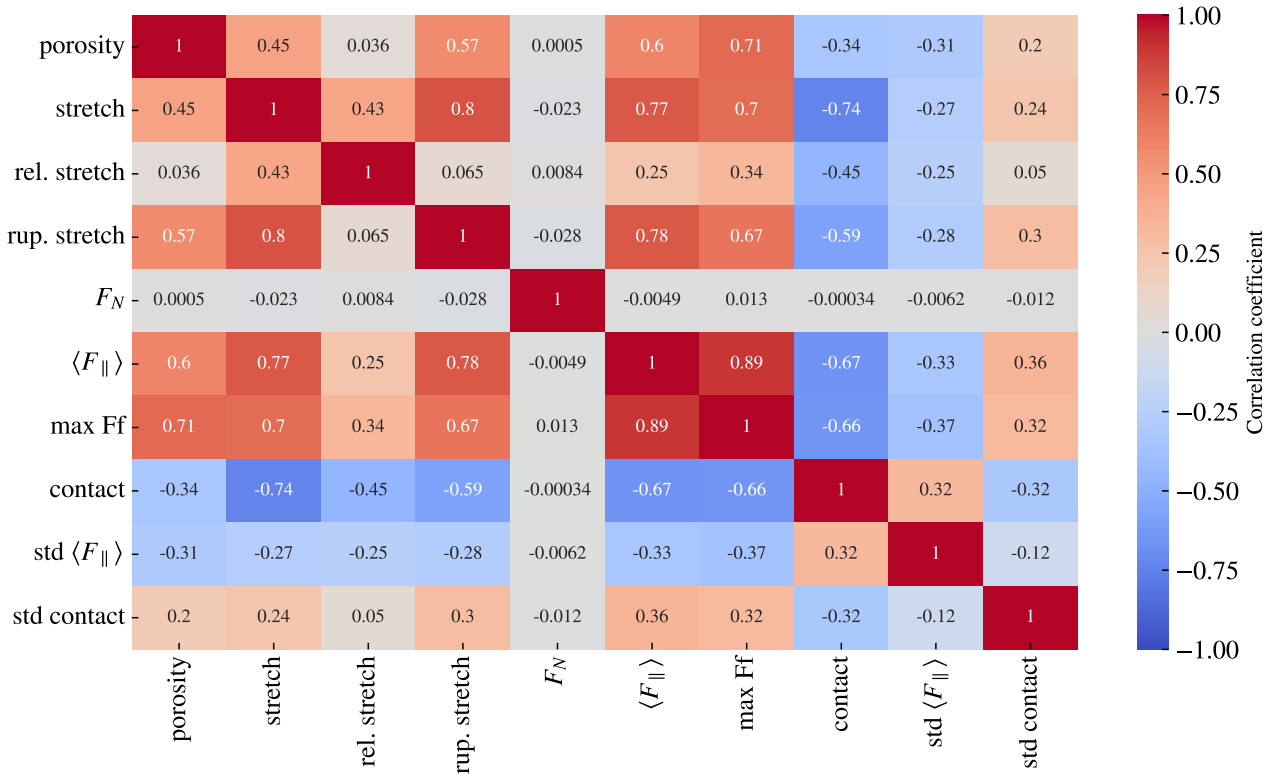


Figure 2.1: Pearson product-moment correlation coefficients for the full datset (see Table 2.1).

From Fig. 2.1 we especially notice that the mean friction force $\langle F_{\parallel} \rangle$ has a significant positive correlation with stretch (0.77) and porosity (0.60). However, the relative stretch, the stretch scaled by the rupture stretch, has a weaker correlation of only 0.25. This indicates that the correlation might be associated with the flexibility of the configurations as these can be taken to higher absolute values of stretch. This is further supported by the fact that the mean friction and the rupture stretch are also strongly positively correlated (0.78). From figure Fig. 2.1 we also observe that the contact is negatively correlated with the mean friction (-0.67) and the stretch value (-0.74) which is consistent with the trend observed in the pilot study in **??** of the contact decreasing with increasing stretch and mean friction. However, we must take note that the correlation coefficient is a measure of the strength and slope of a forced linear fit on the data Do I need as source?. Since we have observed a non-linear trend between friction and stretch (**??**) we should not expect any 100 % correlations Fig. 2.2 shown a visualization of the data (excluding the pilot study configurations) for chosen variable pairs on the axes. This provides a visual clue on some of the correlations and provides a qualitative feeling for the diversity in various slices of parameter space that we eventually are going to base our machine learning model on. We notice that the honeycomb pattern is spanning a significantly larger range of stretch, contact and mean friction.

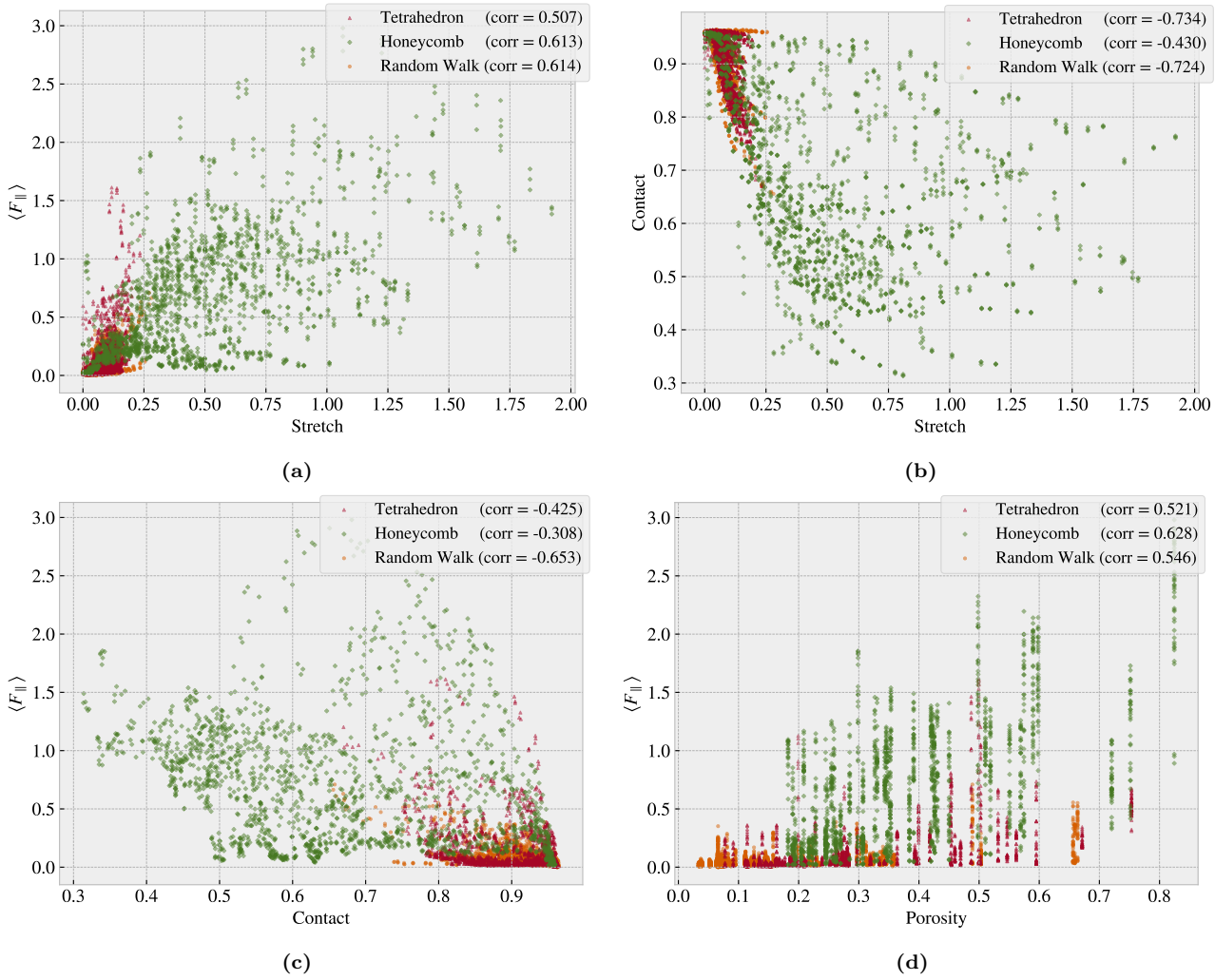


Figure 2.2: Scatter plot of the data sets Tetrahedron, Honeycomb and Random Walk (excluding the pilot study) for various variable combinations in order to visualize some chosen correlations of interest and distributions in the data

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 and stretch. Hence, we will consider this as a main property of interest for our further exploration. However, it is not obvious how one should rigorously quantify this. The friction coefficient is by our definition (**??**) given as the slope of the friction F_f vs. normal force F_N curve. Hence, for two data points $\{(F_{N,1}, F_{f,1}), (F_{N,2}, F_{f,2})\}$, $F_{N,1} < F_{N,2}$ we can evaluate the associated friction coefficient $\mu_{1,2}$ as

$$\mu_{1,2} = \frac{F_{f,2} - F_{f,1}}{F_{N,2} - F_{N,1}} = \frac{\Delta F_f}{\Delta F_N}.$$

In the pilot study, it became clear that the effects of friction under the change of load is negligible in comparison to the effects related to stretch S. Thus, by working under the assumption $F(F_N, S) \sim F(S)$ and a coupling $F_N \propto R \cdot S$ with linear coupling ratio R we get

$$\mu_{1,2}(S_1, S_2) = \frac{\Delta F_f(S_1, S_2)}{R(S_2 - S_1)} \propto \frac{\Delta F_f(S_1, S_2)}{\Delta S}. \tag{2.1}$$

With this reasoning we can in practice exchange F_N with S in the expression for the friction coefficient of our coupled system. This justifies the search for a strong negative slope on the friction vs. stretch curve as it can be related to a negative friction coefficient in our proposed coupled system. The remaining question is 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 ΔS, corresponding to a small denominator in Eq. (2.1), 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 stretch as a better metric. For a given friction vs. stretch curve, we can locate the local maxima and evaluate the difference to the succeeding local minima. The biggest drop will serve 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 friction, max drop, we also consider the minimum, $\min F_{\text{fric}}$, and maximum, $\min F_{\text{fric}}$, friction along with 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 designs categories (Tetrahedron, Honeycomb and Random walk) and the Pilot study are summarized in Table 2.2 with the corresponding stretch profiles and configurations shown in Fig. 2.3 to 2.6 (excluding the pilot study). The stretch profiles for the full dataset are shown in appendix **??**.

From the property comparison in Table 2.2 we see in general that the Honeycomb pattern is superior in terms of the maximum properties which aligns with the pilot study results as well. However, by including more variations of the patterns we find an improvement in the max drop category for both the Tetrahedron pattern ($0.5098 \rightarrow 0.8841$) and the Honeycomb pattern ($0.9674 \rightarrow 1.2785$). The comparison also reveals that the non-cut sheet is still the best candidate for a low friction behavior which indicates that the Kirigami designs cannot be used to further reduce friction in our simulation domain. The Random walk property scores are in general on a comparable order of magnitude which indicates that these randomized patterns have some relevance with respect to the usage as training data.

Table 2.2: Evaluation of the properties of interest for our dataset.

Tetrahedron	Configuration	Stretch	Value [nN]
min F_{fric}	$(3, 9, 4)$	0.0296	0.0067
max F_{fric}	$(5, 3, 1)$	0.1391	1.5875
max ΔF_{fric}	$(5, 3, 1)$	$[0.0239, 0.1391]$	1.5529
max drop	$(5, 3, 1)$	$[0.1391, 0.1999]$	0.8841

Honeycomb	Configuration	Stretch	Value [nN]
min F_{fric}	$(2, 5, 1, 1)$	0.0267	0.0177
max F_{fric}	$(2, 1, 1, 1)$	1.0654	2.8903
max ΔF_{fric}	$(2, 1, 5, 3)$	$[0.0856, 1.4760]$	2.0234
max drop	$(2, 3, 3, 3)$	$[0.5410, 1.0100]$	1.2785

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

Pilot study	Configuration	Stretch	Value [nN]
min F_{fric}	No cut	0.2552	0.0012
max F_{fric}	Honeycomb	0.7279	1.5948
max ΔF_{fric}	Honeycomb	0.7279	1.5325
max drop	Honeycomb	$[0.7279, 1.0463]$	0.9674

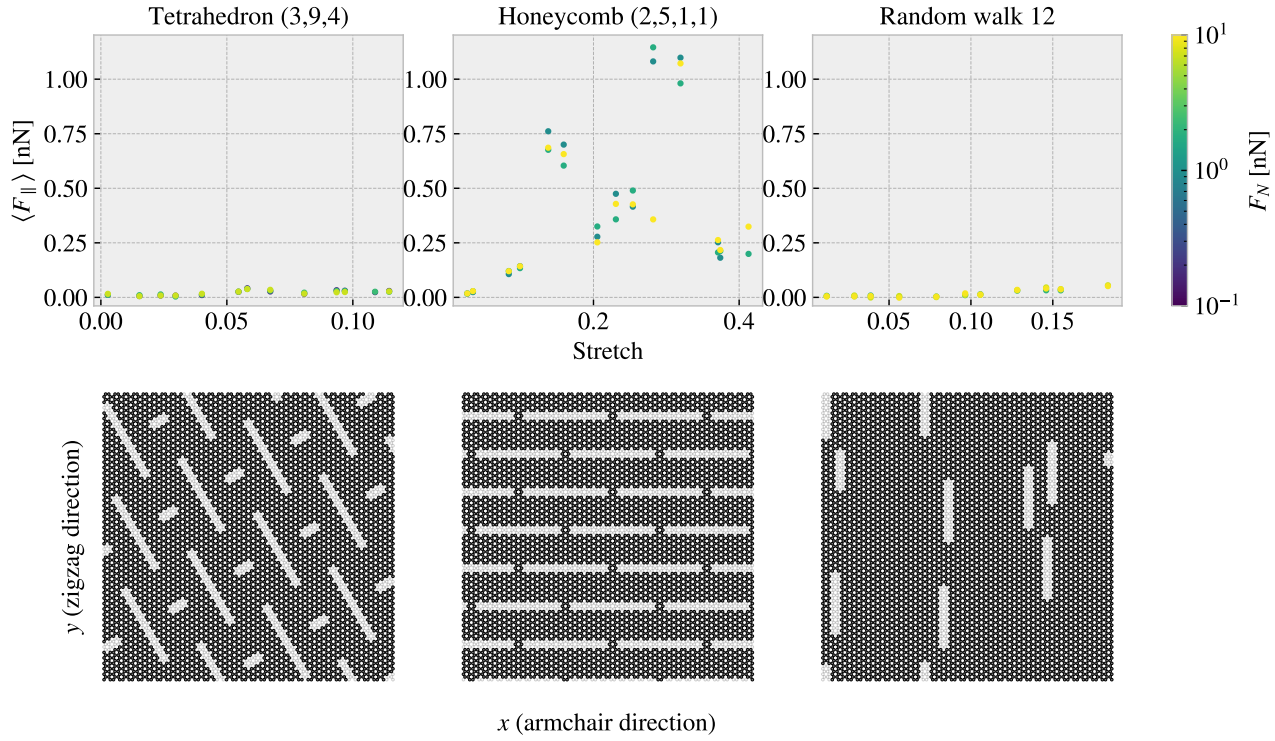


Figure 2.3: Minimum friction: Configurations corresponding to the minimum friction.

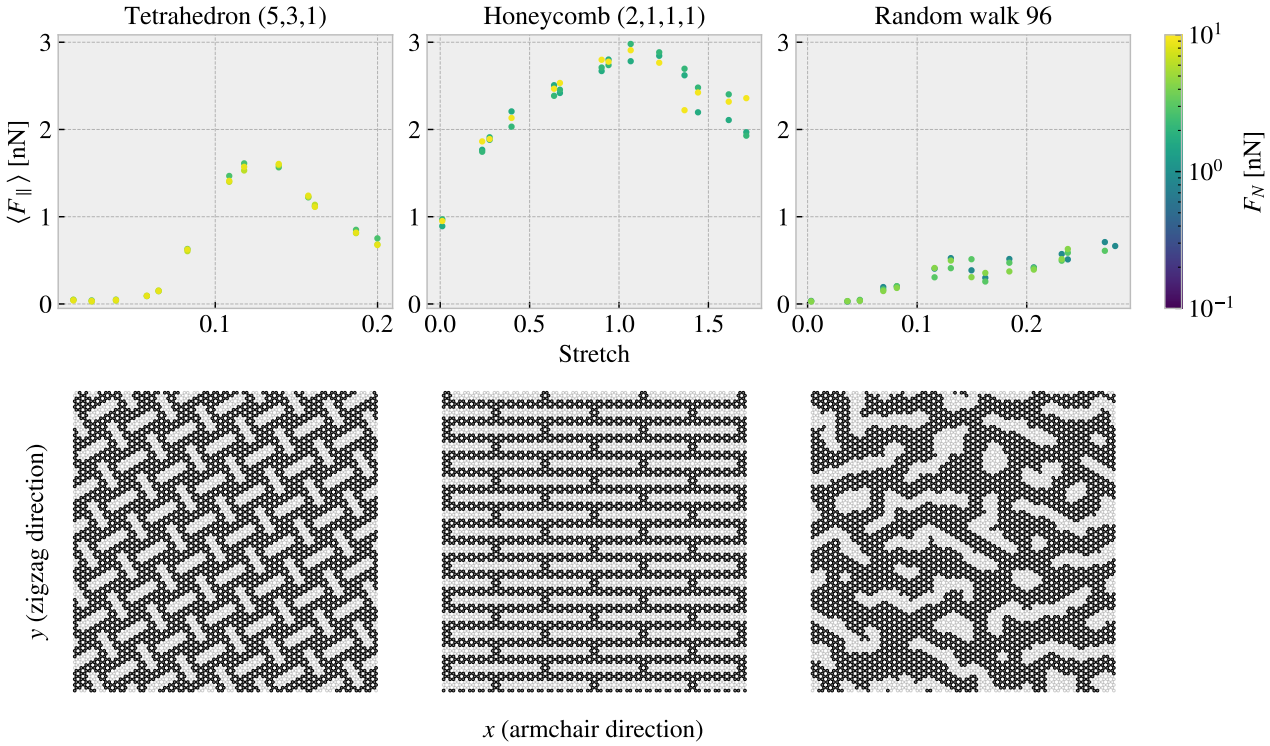


Figure 2.4: Maximum friction: Configurations corresponding to the maximum friction.

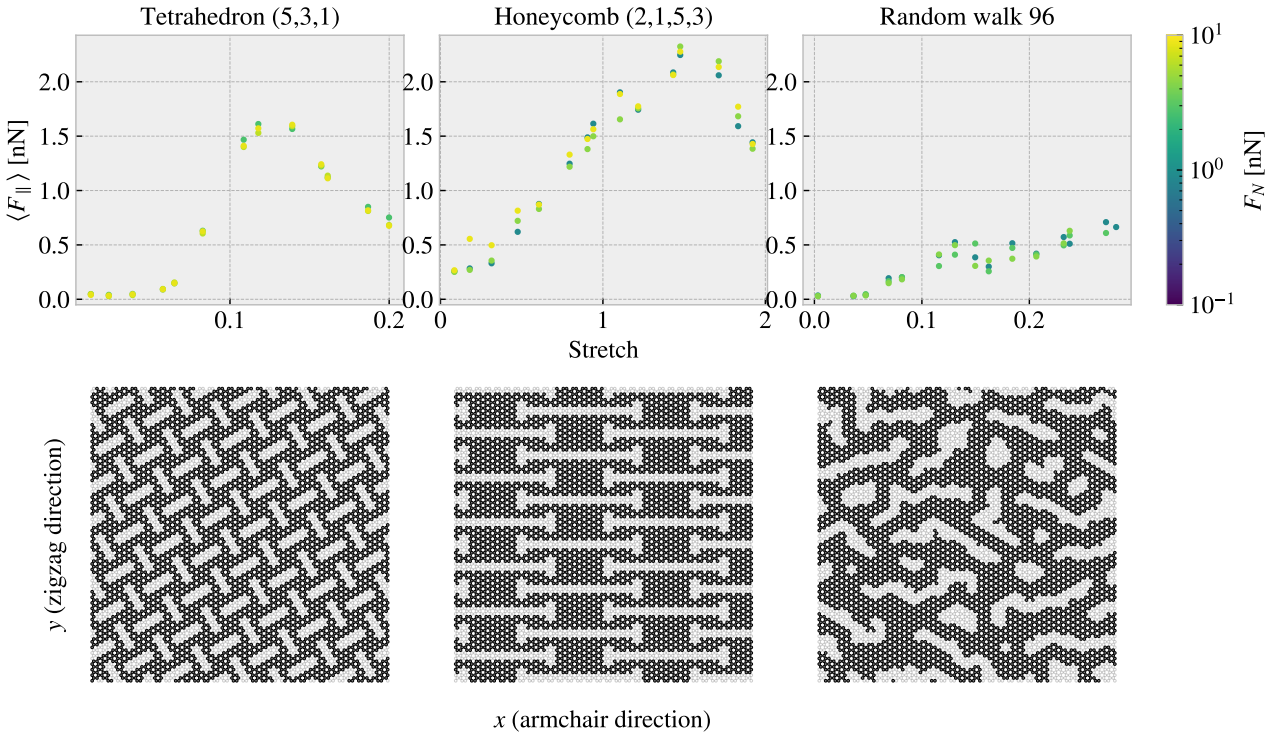


Figure 2.5: Maximum Difference: Configurations corresponding to the biggest difference in friction in the dataset for each pattern.

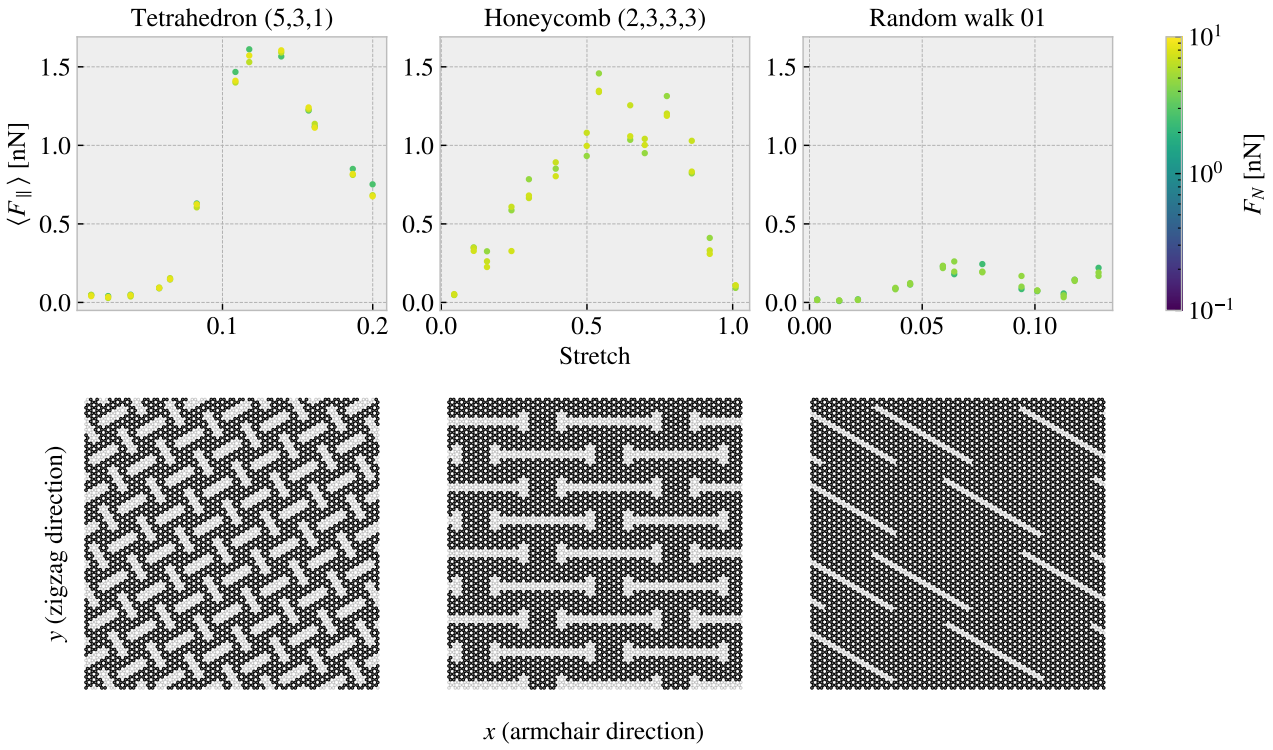


Figure 2.6: Maximum drop: Configuratiosn corresponding to the biggest friction drop in the dataset for each pattern.

2.4 Machine learning

Given the MD-based datset we investigate the possibilities of training a machine learning model to predict the friction behaviour from a given Kirigami configuration, stretch and load.

2.4.1 Architecture

Due to the spatial dependencies in the Kirigami configurations, we use a convolutional neural network (CNN) for our model architecture. 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. 2.7 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. 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 halfs the spatial resolution each time.

- The latter part of the network consists of a fully connected part followed by a ReLU activation. 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 maps the spatial output to the fully connected layer with the number of channels corresponding to the nodes in this layer.

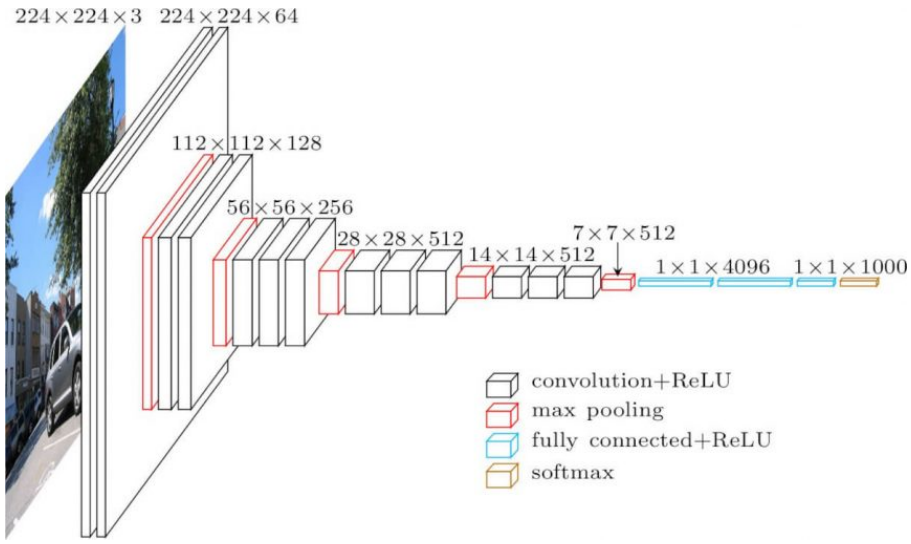


Figure 2.7: VGGNet 16. Source https://neurohive.io/en/popular-networks/vgg16/.

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

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

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 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 with convolutional blocks doubling in channel number for each layer and the latter half with fully connected blocks having the number of nodes decreasing in a reverse pattern. For instance the architecture $S4D8$ will take the form Following this pattern a $(S = 4, D = 8)$ would take the form

$$\text{Input} \rightarrow \underbrace{\overbrace{C4}^{S\,=\,4}\ C8\ C16\ C32\ D32\ D16\ D8\ D4}_{D\,=\,8} \rightarrow \text{Output}.$$

This provides a simple description where S and S 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, stretch of the sheet and applied normal load. While the first is a two-dimensional input the latter are both scalars. This gives rise to two main options for the data structure

1. Expand the scalar values (stretch and load) into 2D matrices of the same size as the Kirigami configuration (copying the scaler value to all positions). 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 first is more appropriate if the configuration should be considered in context with the applied stretch and load, while the latter corresponds to a more independent processing. We implemented both options but it was immediately clear from the test runs that option 1 was producing the most promising result which we settled on.

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 vs. stretch curve with an estimated stopping point as well. However, for some cases, the model can benefit from having additional output variables to adjust for source, and thus we include we include maximum friction, contact, porosity and rupture stretch in the output as well. This also gives us more options for exploring the relationship in the data later on. Notice that we weight the importance of these output variables different in the loss as described in variables differently as described in Sec. 2.4.3.

Removed this part, do you think it has any relevance: Notice that the rupture stretch refers to the value found in the rupture test without load, but as the sheet always ruptures before or just around this point in a loaded state this provides some information for the training to lean on, even though it is in the output state. In principle, we could add a penalty whenever the network predicts the sheet to be attached for stretch values above the rupture stretch, but we found the performance of the rupture prediction to be satisfactory without such penalties.

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 do this across the x-axis as the sheet is translated in a positive y-direction meaning that the reflected version 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 binary values (0: False 1: True). For the scalar values we use the Mean Squared Error (MSE)

$$L_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2,$$

where N is the number of data entries, y are the true output and $\hat{y}$ are the predicted values. For the binary output, we use binary cross entropy

$$L_{\text{BCE}} = -\frac{1}{N}\left[\sum_{i=1}^{N}[t_i \log(p_i) + (1-t_i)\log(1-p_i)]\right],$$

where $t \in \{0, 1\}$ is the truth label. Does this belong in theory entirely? I do introduce it there but I guess I have to mention the choice here still. 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. We found that the results varied little and concluded that the training is not very sensitive to this choice, for which we stuck with the values defined above.

2.4.4 Hypertuning

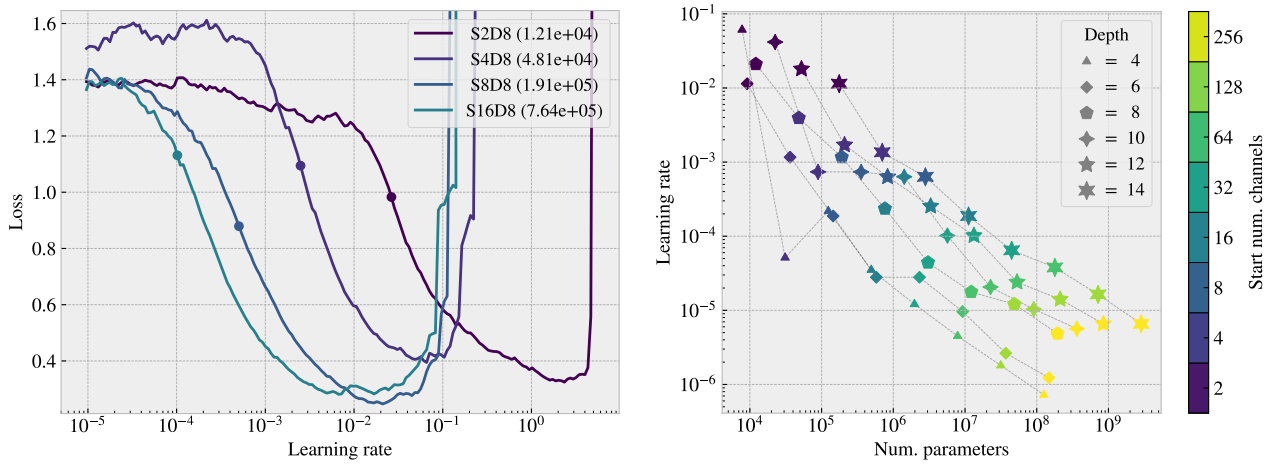
For the hypertuning we focus on architecture complexity, learning rate, momentum and weight decay. Thus the non-chaning elements are the use of the ADAM optimizer with the initial default values of $beta_1 = 0.9, \beta_2 = 0.999$ and zero weight decay (we will change momentum β_1 and weight decay). We use a batch size of 32 and train the model for a maximum of 1000 epochs, but we save the best model during training based on the lowest validation loss.

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 the momentum.

We consider first the architectures 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, increasing the the learning rate until the training loss diverges. The suggested learning rate is then determined as the point for which the training 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 training 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. 2.8 for variours model architectures. 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.



(a) **(b)**

Figure 2.8: Learning rate range test for various model complexities. We increase the learning rate exponentially from nume-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 learning rate. The examplatory 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. 2.8 we perform a grid search over the corresponding S and D parameters. We evaluate both the loss and the mean friction R_2 score for the validation data which is shown in Fig. 2.9 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 maximum friction drop for the Tetrahedron and Honeycomb pattern respectively. This is done as a way of evaluating the performance on the non-linear stretch curves which we immediately found to be the more difficult patterns to capture. The selected evaluation is shown in Fig. 2.10. Note that these patterns are already

a part of the full dataset and thus the data points related to these patterns 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. 2.9.

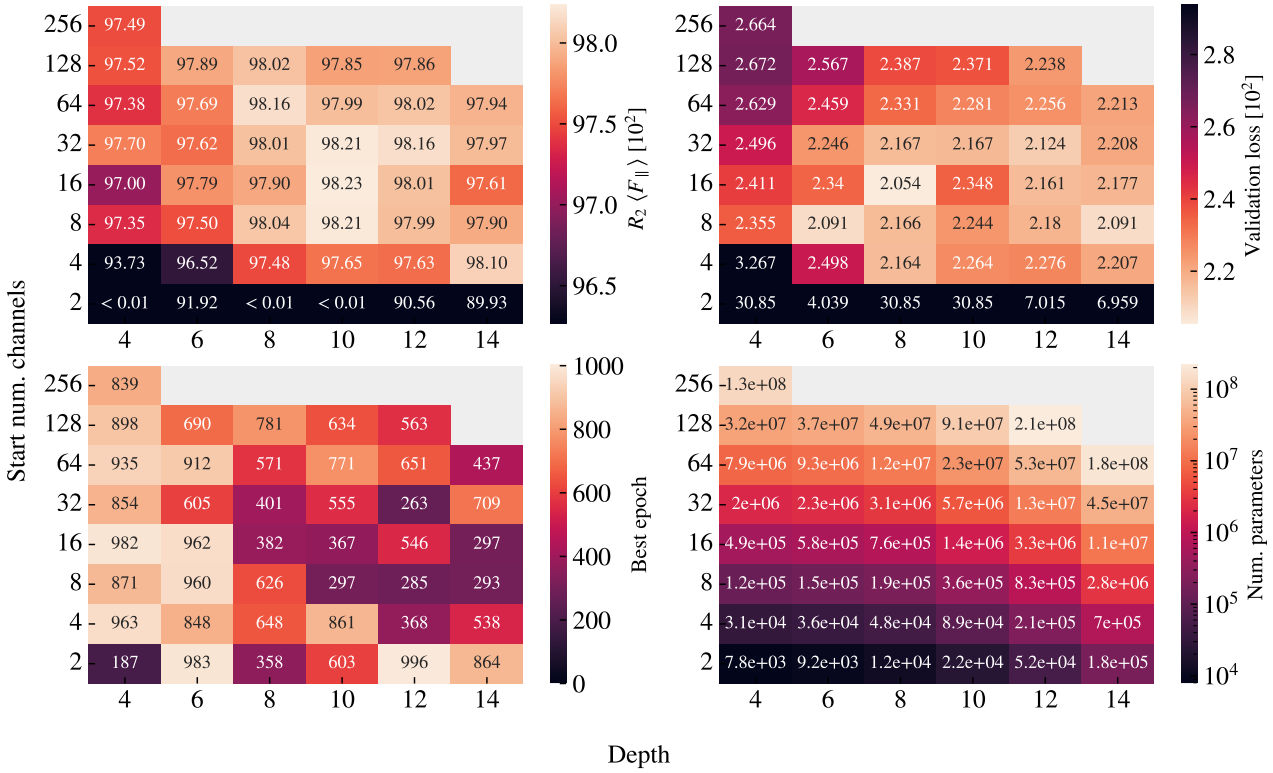


Figure 2.9: Architecture search.

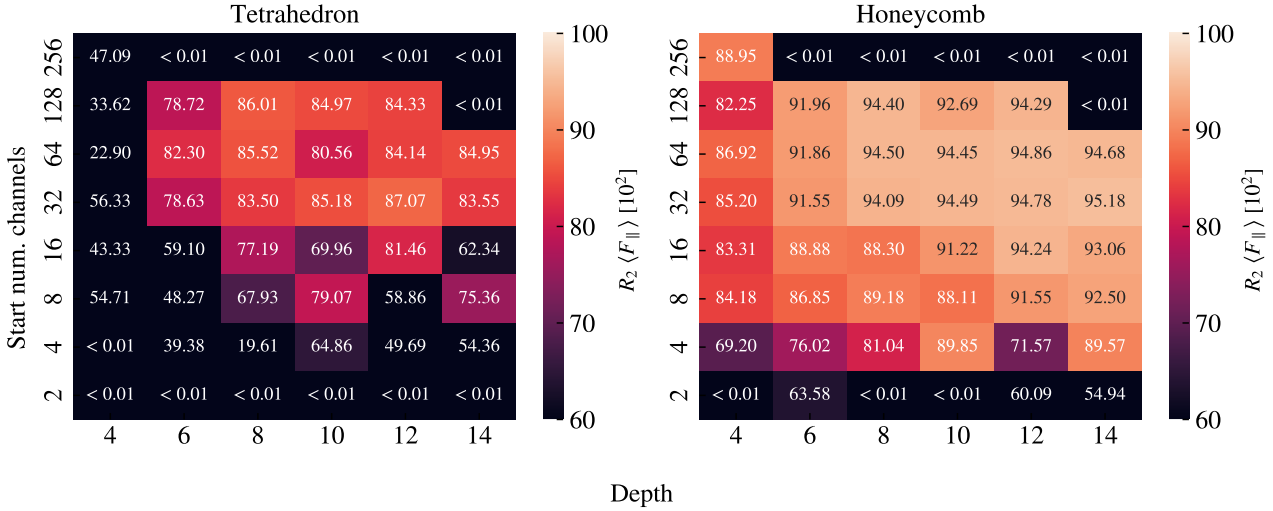
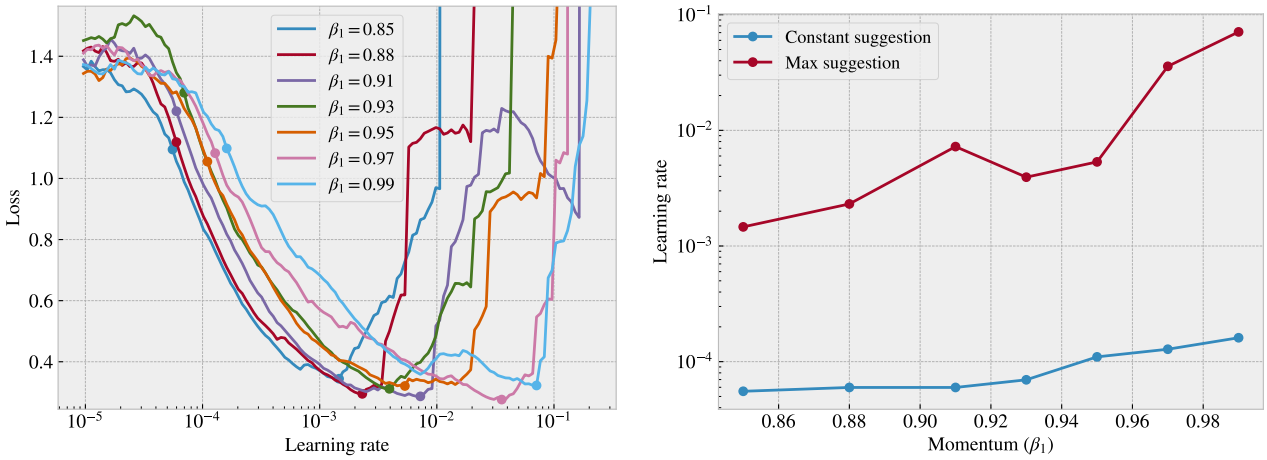


Figure 2.10: Selected pseudo validation set. Fix the missing grey fields in the top which are replaced by ¡ 0.01.

From the validation scores in Fig. 2.9 we find that models S(8-32)D(8-12) to give the best performance. When looking at the best epoch we find that models of low depth result in a later best epoch which is compatible with underfitting. As the depth is increased we find more models with a lower best epoch, in the range $\sim [300, 600]$,

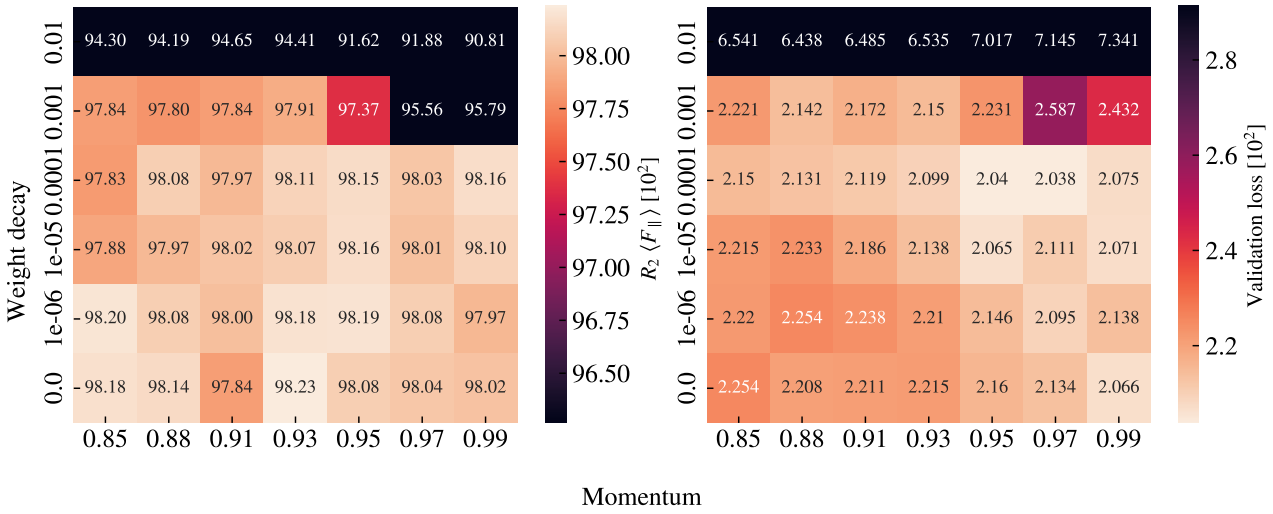
which on the other hand suggest cases of overfitting. Since our training stores the best model during training, we do not have to worry too much about overfitting, but we can take this transition from underfitting to overfitting as a sign that our search is conducted in an appropriate complexity range. When consulting the evaluation on the selected set in Fig. 2.10 we notice in general that we get 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 is a clear indication that these configurations are more challeging to predict. While the peak R_2 value for the validation score was found for the S16D10 model we see a slight preference for more complexity in the model with regard to the selected test. In the Tetrahedron selected set grid search, we find the best model to be S32D12, with a R_2 score of $\sim 87\%$. This model choice is more or less compatible with the overall performance as this is among the top candidates for the R_2 score and loss in Fig. 2.9 and the R_2 score for the Honeycomb pattern in Fig. 2.10 as well. Hence, we settle on this architecture.

Next, we consider momentum m and weight decays wd in the range $m \in [0.85, 0.99]$ and $wd \in [0, 1e-2]$. An increased momentum is expected to decrease the appropiate learning rate, and thus we perform a new learning rate range test to determine a suitable choice for the learning rate for each momentum choice. 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. We then decreases towards a final minimum being a factor $1e-4$ of the maximum bound during the remaining 70% of training. The increase and decrease is 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 of diverging. We find that the minimum point on the loss curve is as suitable choice that approaches the diverging point without getting to close and causing diverging learing. The learning rate range test for momentum is shown in Fig. 2.11. Using the results for the momentum learning rate range test 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 friciton mean R_2 score for the selected set of Tetrahedron and Honeycomb patterns. This is shown for the constant learning rate scheme in Fig. 2.12 and for the cyclic scheme in Fig. 2.13.

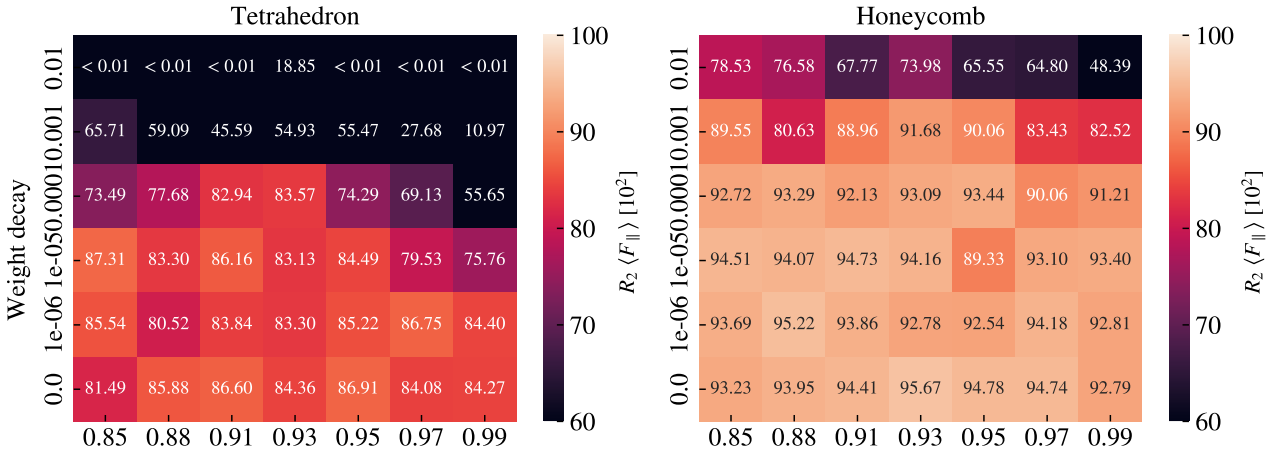


(a) **(b)**

Figure 2.11: Momentum learning rate range tets

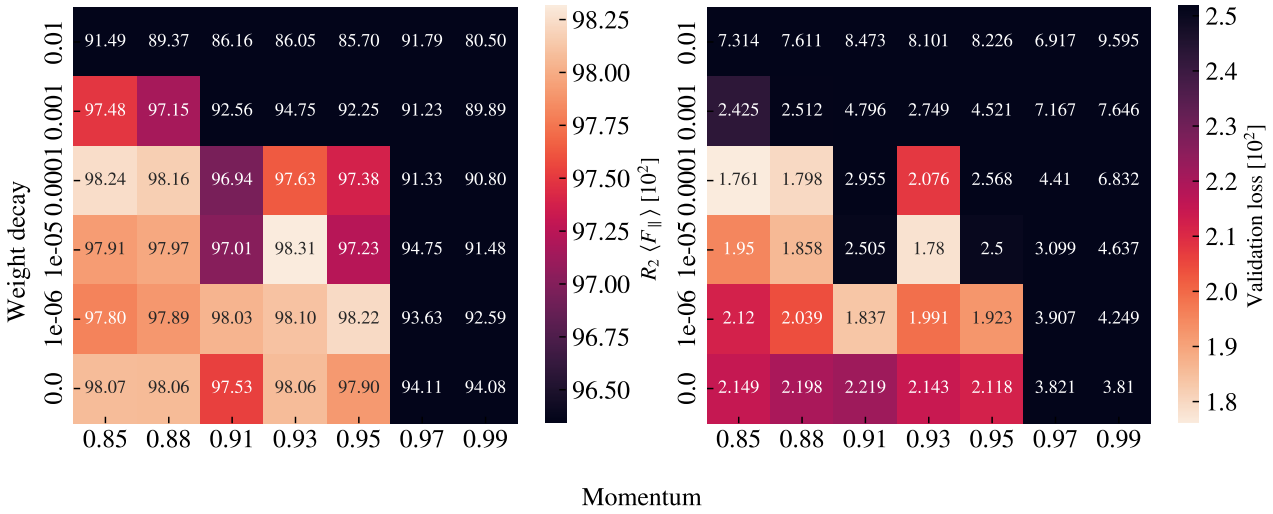


(a) Performance

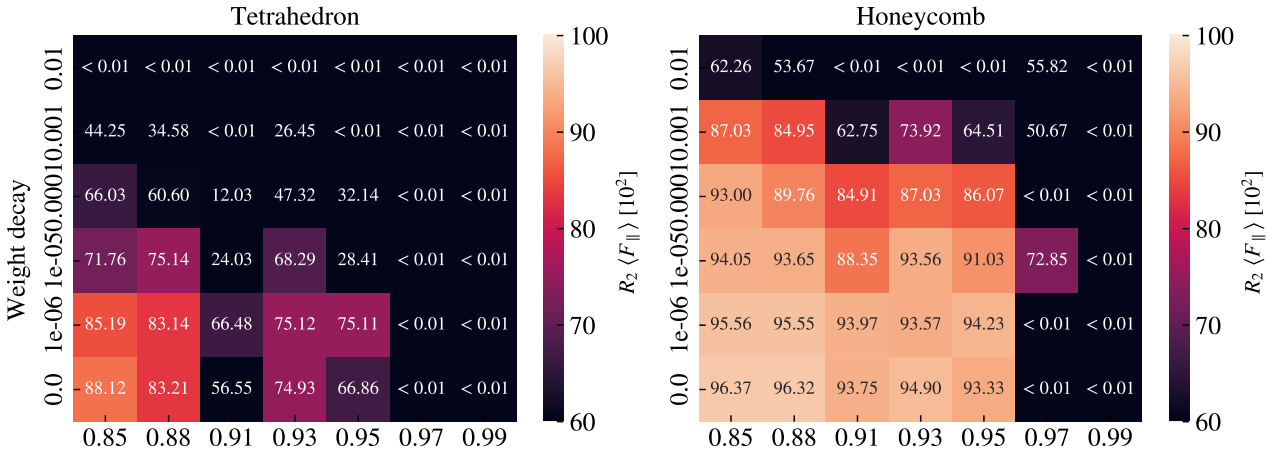


(b) Compare

Figure 2.12: Constant learing rate and momentum scheme



(a) Performance



(b) Compare

Figure 2.13: Cyclic learing rate and momentum scheme

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 are able to improve the performance metrics 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 notice that these scores a taking for their individual optimial hypersettings. The comparison among best scores are summarized in Table 2.3. In general the constant scheme shows reasonble stable results for all momentum settings $m \in [0.85, 0.99]$ in combination with a low weight decay $wd \leq 10^{-4}$. For the cyclic scheme the performance peaks towards a low momentum $m \leq 0.93$ and low weight decay $wd \leq 10^{-4}$. Looking at the summary in Table 2.3 we see that the cyclic scheme is able to produce a high score among all four performance metrics, however since these do not share common hyperparameters we need to make a choice here.

Table 2.3: Momentum and weight decay grid search using S32D12 model.

		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

Look at overfitting via training history.

2.4.5 Final model

From the hypertuning study we choose the S32D12 model trained by a cylic training scheme with momentum 0.85 and weight decay 0. The main performance metrics are shown in Table 2.4. Since the porosity is a number between 0 and 1 we can interpret the absolute error as the percentage error similar to the relative error for the rupture stretch. The rupture stretch is generally within a 13 % margin, but this number might especially high due to some low stretch rupture cases in the dataset. The stretch curves for mean friction, max friction and contact is shown in Fig. 2.14 for the Tetrahedron (7, 5, 1) and Honeycomb (2,2,1,5) used in the pilot study. This gives a visual interpretation of how well the fits actually are for a given R_2 scores, and we notice that a R_2 score above 0.98 is certainly visually promising for capturing the non-linear in the data.

Table 2.4: Mean values are used over different configurations.

	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. Stretch	Rupture
Validation	2.1488	98.067	93.558	94.598	02.325	12.958	96.102
Tetrahedron	4.0328	88.662	85.836	64.683	01.207	05.880	99.762
Honeycomb	8.6867	96.627	89.696	97.171	01.040	01.483	99.111

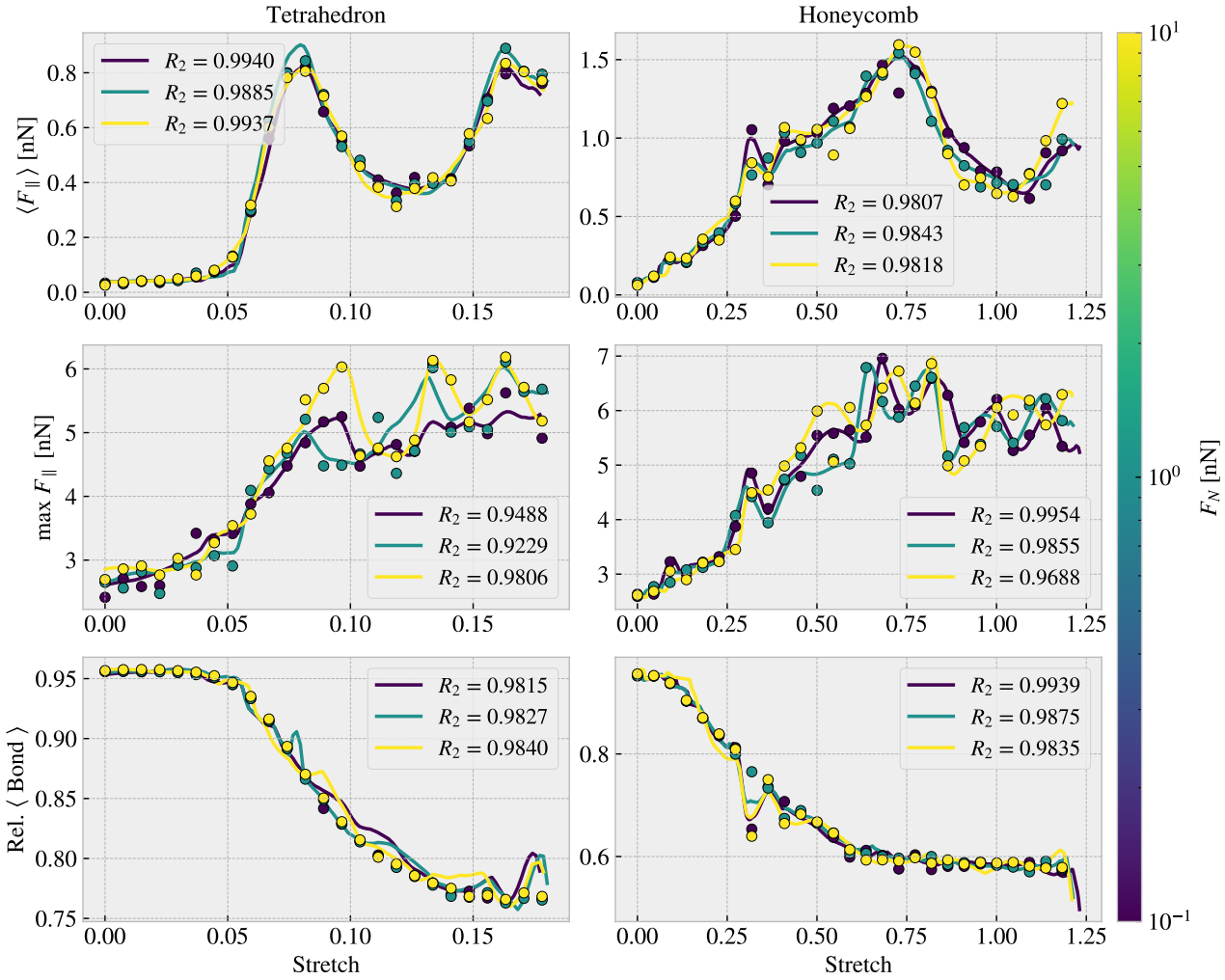


Figure 2.14: With 10^3 points in the stretch range [0, 1.5] and stopping after first rupture True prediction.

Having a trained model, we evaluate the performance for the top ranking within the properties of interest. That is, we go through all the configurations in the dataset for the Tetrahedron (Table 2.5), Honeycomb (Table 2.6) and Random Walk (Table 2.7) patterns seperately and rank the configurations in each property category. This is then compared to the actual ranking in the dataset. Generally, we find that the ML performs rather well in the ranking of the max, max difference and max drop property, but it is deviating a bit more for the minimum friction property. This is most likely due to the fact that the difference in minimum value is quite small compared to that in the max categories. The ML model gives stable prediction for the max-categories for the Tetrahedron and Honeycomb pattern, but struggles a bit more for the Random Walk. Looking at the values for the top candidates in the max-types properties we see that it is generally within a ~ 0.2 nN range which is promising.

Table 2.5: Tetrahedon

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

We use the ML model perform a search through new configurations which optimize our properties of interest. We approach this 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.

2. Using the genetic algoritm method we pertubate 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 to create an extended dataset for our search. For the Tetrahedron and Honeycomb patterns, the increment of the parameters will eventually lead to the main structures becoming so large they do not really fit on the sheet anymore. 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. We use a random reference position and regenerate each unique parameter 10 times to explore translational effects. This gives in total 135k configurations for the Tetrahedron pattern and 2025k 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 an uniform (U) or logarithmic uniform (LU) distribution as follows.

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

Notice that we use discrete distribution for the parameters requiring integers. For the binary parameters *Connection*, *Avoid unvalid*, *RN6* and *Grid start* we simply set the values by 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 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 evaluation of the configurations we use a normal load of 5 nN and generate a stretch curve in the domain 0–200 % using 100 evenly spacd points. top candidate results for each property are shown in Table 2.8 including a comparison to the dataset top candidates originally shown in Table 2.2. The random walk top five candidates are visualized in Fig. 2.16.

First of all, the search gives a rather consistent result regarding the minimization of friction where the top candidates all share the same feature of being sparsely cut. For the Random Walk we see this clearly inFig. 2.16, while for the Tetrahedon 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 $1.5\%, 5.6\%, 1.6\%$ for the Tetrahedron, Honeycomb and Random walk respectively. These results point towards the fact that the kirigami modification does not immediately give rise to any lowering of friction (within our MD parameter domain).

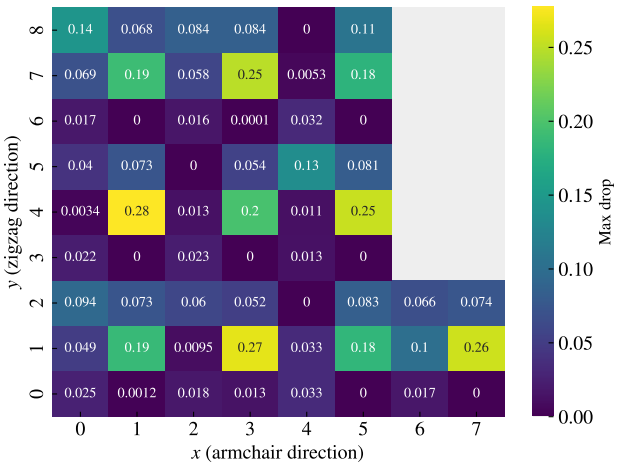
Note that the full ranking did not strictly favor the least amount of atoms removed, but considering that the relative error is high for this domain we can not expect this kind of precision. The fact that the top candidates all take negative values also shows that the model is not reliable in this domain. By asking for the lowest friction we essentially find the weak spots, sparsely populated points in vector space, which can lead to unphysical predictions. This problem should most likely be resolved through an extension of the dataset and perhaps by applying a physical constraint to the model regarding negative values.

Among the remaining properties, we find competing values for the Honeycomb and Random walk classes only. When taking a closer look to the ranking for each property it became apparent the predictions are highly sensitive to the reference position parameters used for the Tetrahedron and Honeycomb pattern. Since we repeated each parameter 10 times with random reference positions, we initially expected to get a ranking in sets of 10. However, the ranking only shows contiguous appearing sets in the range 1–5. Hence we investigate this sensitivity further by evaluating the scores for a systematic change of the reference position for selected configurations. We generally found the mac drop parameter to give the highest variation and thus we show 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. 2.15. It becomes evident that the predictions vary drastically with translation of these patterns. The emerging question is then whether this is actually grounded in a physical phenomenon or simply a deficiency in the ML. Even though the patterns are periodic in the x-y-plane, by the number of center elements represented by the shown squares in Fig. 2.15, 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 look back at theory and maybe source, and thus for a sheet where the atoms sitting on the $\pm x$ free sides constitutes 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. We earmark this suggestion for another study. When considering some of the stretch curves we also find that the prediction of the rupture point makes a crucial impact. As the rupture was often predicted on a descending part of the curve any variation to the rupture point will affect the max drop property quite significantly.

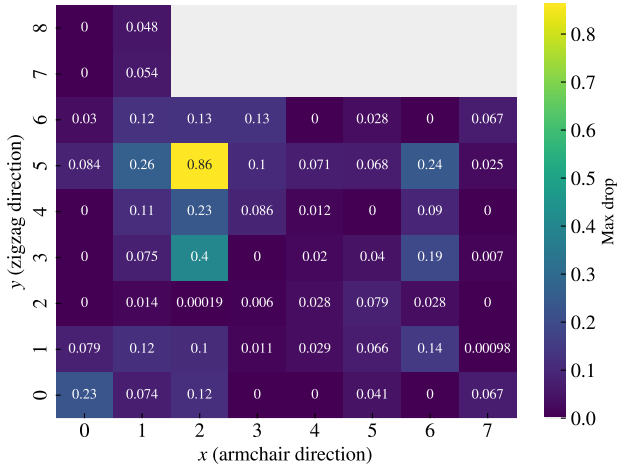
Table 2.8: Pattern search. The values are in units nN.

	Search			Data		
Scores	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

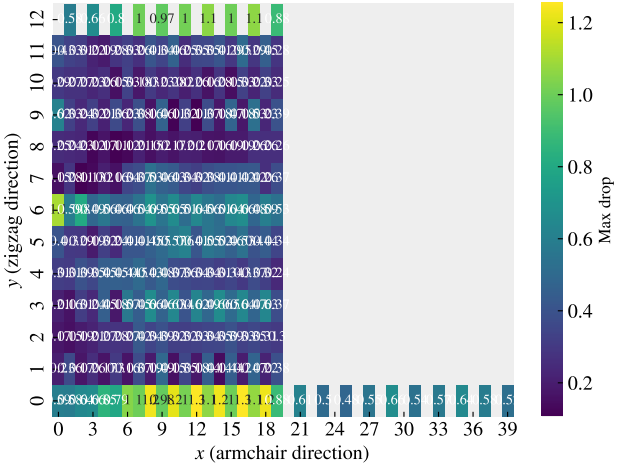
Configs.	Tetrahedron	Honeycomb	Random walk	Tetrahedron	Honeycomb	Random walk
min F_{fric}	$(13, 11, 14)$	$(14, 25, 7, 19)$	No naming	$(3, 9, 4)$	$(2, 5, 1, 1)$	12
max F_{fric}	$(1, 3, 1)$	$(2, 1, 1, 1)$	No naming	$(5, 3, 1)$	$(2, 1, 1, 1)$	96
max ΔF_{fric}	$(1, 3, 1)$	$(2, 1, 1, 1)$	No naming	$(5, 3, 1)$	$(2, 1, 5, 3)$	96
max drop	$(1, 7, 1)$	$(3, 3, 5, 3)$	No naming	$(5, 3, 1)$	$(2, 3, 3, 3)$	01



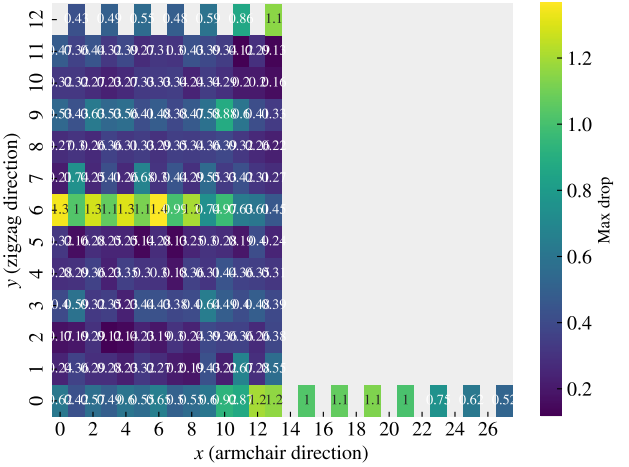
(a) Tetrahedron $(1, 7, 1)$ (60). Std = 0.08, Rel. Std = 1.13



(b) Tetrahedron $(5, 3, 1)$ (60). Std = 0.13, Rel. Std = 1.61



(c) Honeycomb $(3, 3, 5, 3)$ (260). Std = 0.25, Rel. Std = 0.58



(d) Honeycomb $(2, 3, 3, 3)$. (182) Std = 0.27, Rel. Std = 0.60

Figure 2.15: CAPTION

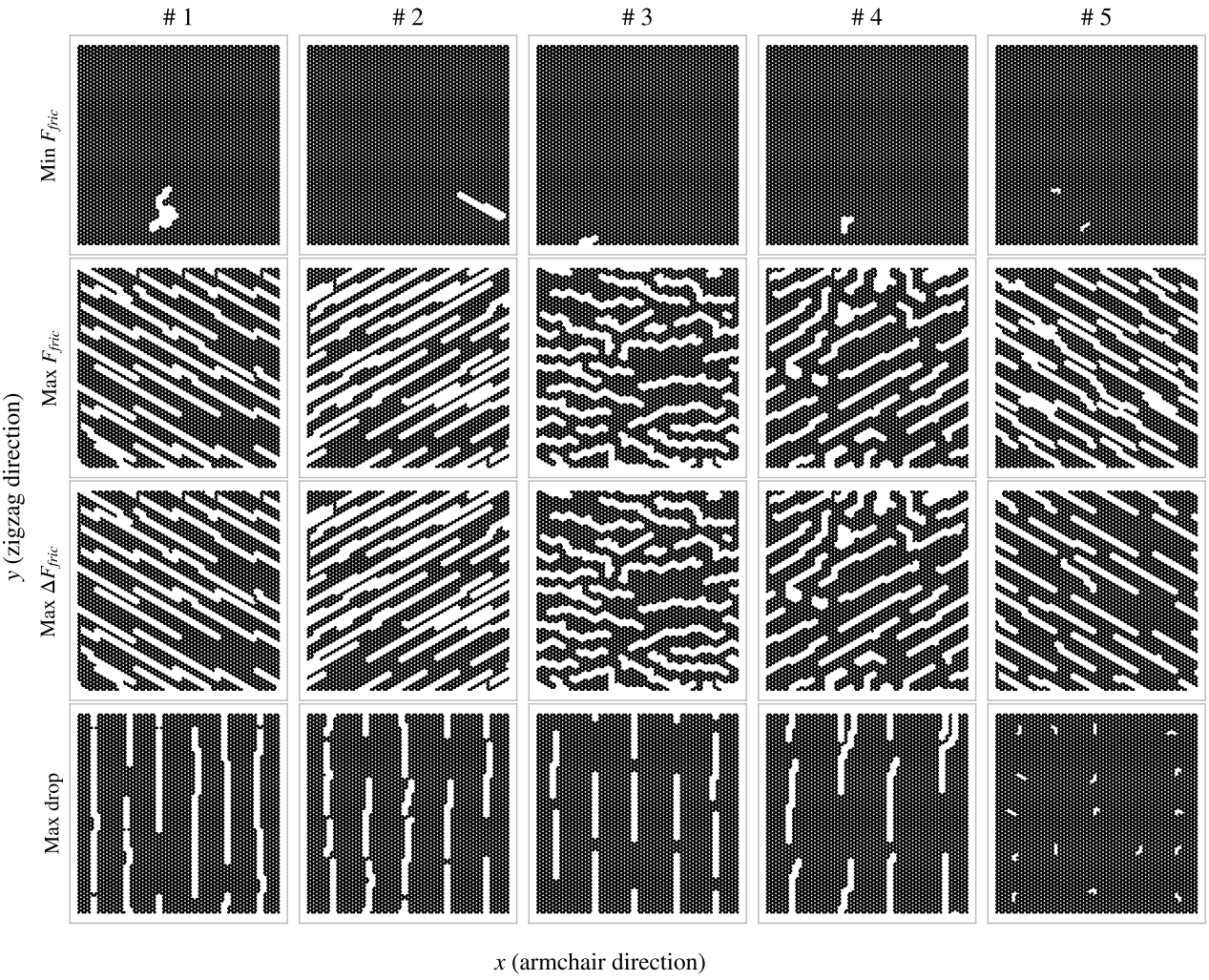


Figure 2.16: RW search top results.

2.5.2 Genetic algorithm search

The second approach to an accelerated search is through the genetic algorithm. So far we have concluded that a minimization of the friction, with our system specifications and model predictions, is not promising. Hence, we discard this property for further study. We have also seen that the maximum style property often shares similar top candidates, and thus we choose to only investigate the max drop property, associated with the aim of creating a negative friction coefficient. From the extended search, we have three candidates within the Tetrahedron, Honeycomb and Random walk style. We use these three candidates as the basis for an genetic agorithm search. We generate a population of 100 configurations using the settings associated with the top candidates. We run the search for 50 generations as we did not see much difference for any longer extensions. The Tetrahedron and Honeycomb search did not give any improvements as the top configuration as the first generations was not improved on even through the average score was rising. This was more or less the case for the random walk as well with only a single new candidate giving a score of 0.300 nN for the drop property. The fact that starting from an existing design did not give any useful results we attempted to start from a population of random noise as well. We did with mixed porosities, even parts of $\{0.01, 0.05, 0.1, 0.2, 0.3\}$, and two for a porosity of 0.25 and 0.5 respectively. This time the algorithm improved the top candidate throughout, but the actual scores were disappointing. For the mixed porosity start, we found a top score of 0.299 nN. The top candidates did not seem to carry any higher order patterns. To the human eye they still looked like random noise. By the use of the gradient cam method not sure of name yet we investigate for any noticable patterns in the model. This is shown for the mixed porosity top candidate in Fig. 2.17 which can be compared to the top search candidates in the

drop category in figure Fig. 2.18, Fig. 2.19 and Fig. 2.20. While looking a such gradient visulizations can be mesmerizing they do not immediately reveal more about the underlying mechanisms. The "attention" of the model often matches well with the placement of the cuts, but it also reveals some frames where it seems consider the edge quite heavily. This especially relates to to the top and bottom edge which corresponds to the front and back of the sheet. Since these are also connected to the pull block it is really not an edge and thus it is a bit surprising if these should be of extra importance. Given the limited time resources we can follow up on these questions.

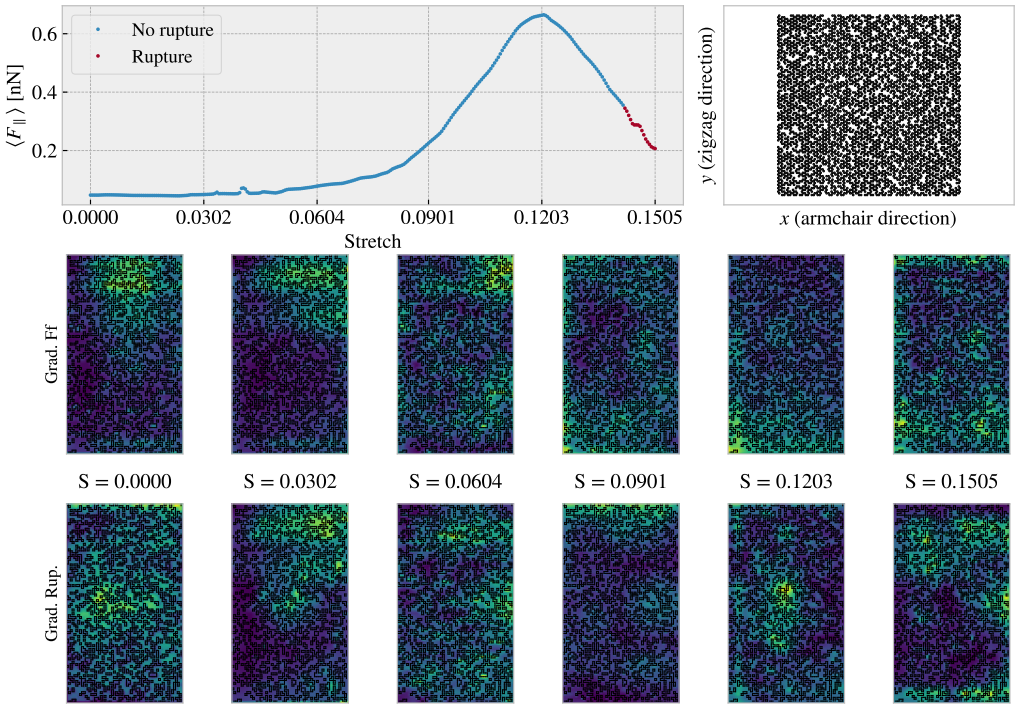


Figure 2.17: $p \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$.

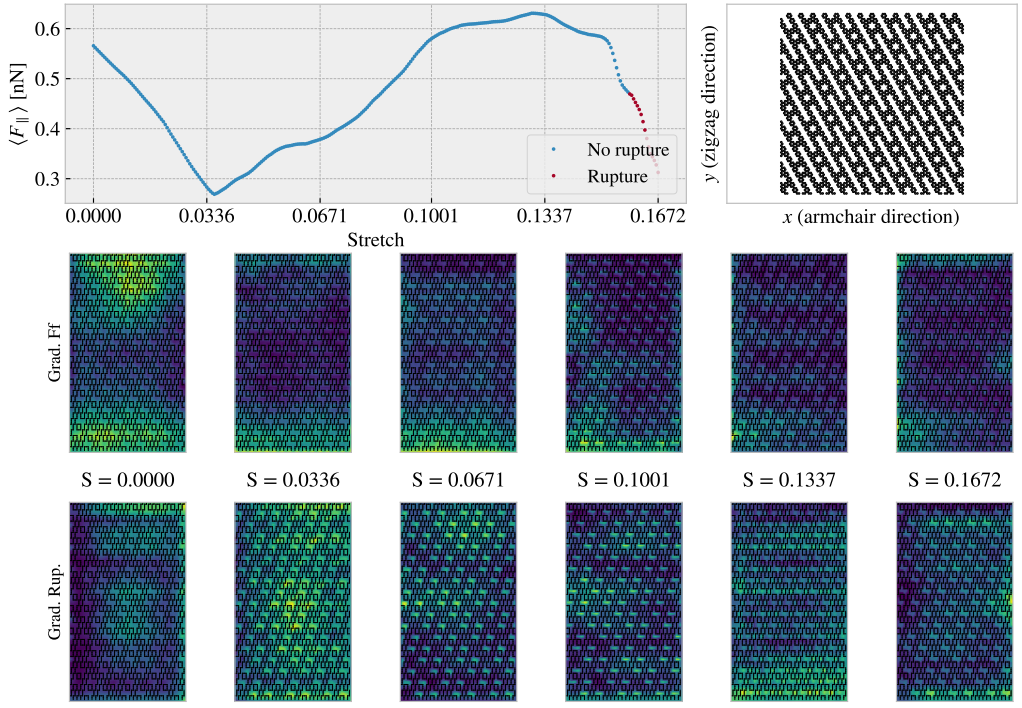


Figure 2.18: Tetrahedron $(1, 7, 1)$, ref $= (1, 4)$

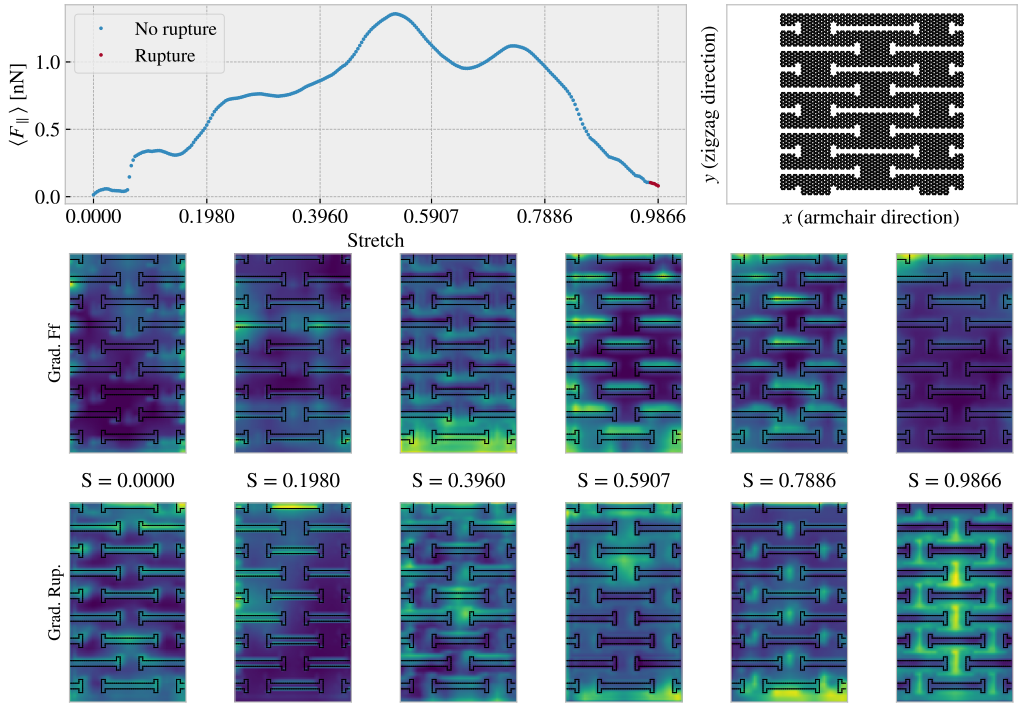


Figure 2.19: Honeycomb $(3, 3, 5, 3)$, ref $= (12, 0)$

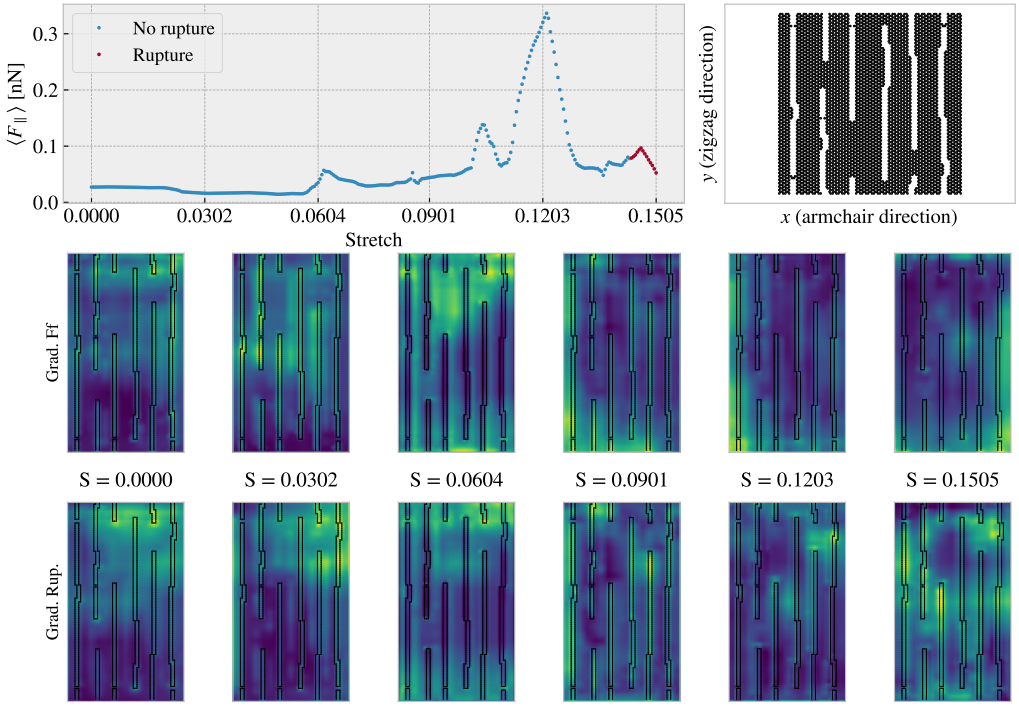


Figure 2.20: RW.

Appendices

Appendix A

Appendix A

Appendix A

Appendix B

Appendix B

Appendix C

Bibliography

[1]E. Gnecco and E. Meyer, *Elements of friction theory and nanotribology* (Cambridge University Press, 2015).

[2]Bhusnan, "Introduction", in *Introduction to tribology* (John Wiley & Sons, Ltd, 2013) Chap. 1, 1–?

[3]H.-J. Kim and D.-E. Kim, "Nano-scale friction: a review", International Journal of Precision Engineering and Manufacturing **10**, 141–151 (2009).

[4]K. Holmberg and A. Erdemir, "Influence of tribology on global energy consumption, costs and emissions", Friction **5**, 263–284 (2017).

[5]B. Bhushan, "Gecko feet: natural hairy attachment systems for smart adhesion – mechanism, modeling and development of bio-inspired materials", in *Nanotribology and nanomechanics: an introduction* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2008), pp. 1073–1134.

[6]P. Z. Hanakata, E. D. Cubuk, D. K. Campbell, and H. S. Park, "Accelerated search and design of stretchable graphene kirigami using machine learning", Phys. Rev. Lett. **121**, 255304 (2018).

[7]P. Z. Hanakata, E. D. Cubuk, D. K. Campbell, and H. S. Park, "Forward and inverse design of kirigami via supervised autoencoder", Phys. Rev. Res. **2**, 042006 (2020).

[8]L.-K. Wan, Y.-X. Xue, J.-W. Jiang, and H. S. Park, "Machine learning accelerated search of the strongest graphene/h-bn interface with designed fracture properties", Journal of Applied Physics **133**, 024302 (2023).

[9]Y. Mao, Q. He, and X. Zhao, "Designing complex architectured materials with generative adversarial networks", Science Advances **6**, eaaz4169 (2020).

[10]Z. Yang, C.-H. Yu, and M. J. Buehler, "Deep learning model to predict complex stress and strain fields in hierarchical composites", Science Advances **7**, eabd7416 (2021).

[11]A. E. Forte, P. Z. Hanakata, L. Jin, E. Zari, A. Zareei, M. C. Fernandes, L. Sumner, J. Alvarez, and K. Bertoldi, "Inverse design of inflatable soft membranes through machine learning", Advanced Functional Materials **32**, 2111610 (2022).

[12]S. Chen, J. Chen, X. Zhang, Z.-Y. Li, and J. Li, "Kirigami/origami: unfolding the new regime of advanced 3D microfabrication/nanofabrication with "folding"", Light: Science & Applications **9**, 75 (2020).

[13]Z. Deng, A. Smolyanitsky, Q. Li, X.-Q. Feng, and R. J. Cannara, "Adhesion-dependent negative friction coefficient on chemically modified graphite at the nanoscale", Nature Materials **11**, 1032–1037 (2012).

[14]B. Liu, J. Wang, S. Zhao, C. Qu, Y. Liu, L. Ma, Z. Zhang, K. Liu, Q. Zheng, and M. Ma, "Negative friction coefficient in microscale graphite/mica layered heterojunctions", Science Advances **6**, eaaz6787 (2020).

[15]D. Mandelli, W. Ouyang, O. Hod, and M. Urbakh, "Negative friction coefficients in superlubric graphite–hexagonal boron nitride heterojunctions", Phys. Rev. Lett. **122**, 076102 (2019).

[16]R. W. Liefferink, B. Weber, C. Coulais, and D. Bonn, "Geometric control of sliding friction", Extreme Mechanics Letters **49**, 101475 (2021).